

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-40079

(43) 公開日 平成10年 (1998) 2月13日

(51) Int. Cl. °	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 7/52	3 1 0		G 0 6 F 7/52 3 1 0	F
7/50			7/50	N

審査請求 未請求 請求項の数10 O L (全 29 頁)

(21) 出願番号 特願平8-194920

(22) 出願日 平成8年 (1996) 7月24日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 辻橋 良樹

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

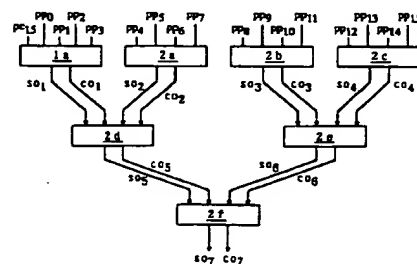
(74) 代理人 弁理士 吉田 茂明 (外2名)

(54) 【発明の名称】 ツリー回路

(57) 【要約】

【課題】 回路動作の並列度を向上させ、以て乗算器の高速化を図る。

【解決手段】 Tree回路の1段目において、4入力2出力加算ブロック2 a～2 cと共に拡張型4入力2出力加算ブロック1 aを設ける。また、2段目において4入力2出力加算ブロック2 d、2 eを、三段目において4入力2出力加算ブロック2 fをそれぞれ設ける。同一段の各加算ブロックの入力信号の到達時刻を揃え、Tree回路のクリティカルパスの論理段数が削減される。



## 【特許請求の範囲】

【請求項1】 ブースのアルゴリズムに従って生成された複数の部分積に基づいて累進的な加算を行うことにより、個数が減減する中間和を生成しつつ、一对の最終中間和を出力するツリー回路であって、複数のビット幅のデータの複数の加算して一对の前記中間和を出力する通常型加算ブロックと、複数のビット幅のデータの複数の、1ビット幅のデータとを加算して一对の前記中間和を出力する拡張型加算ブロックとを備えるツリー回路。

【請求項2】 各々の前記部分積は、複数のビットの第1要素と1ビットの第2要素との和に対して自身の最下位ビット位置を示す位取り部が乗じられたものとして表され、

前記拡張型加算ブロックは、前記複数の部分積を入力し、自身に入力される前記複数の部分積以外の前記複数の部分積に属する第2要素の一つを更に入力する、請求項1記載のツリー回路。

【請求項3】 前記拡張型加算ブロックに入力される前記第2要素は、前記複数の部分積のうち前記位取り部が最も大きな部分積に属する、請求項2記載のツリー回路。

【請求項4】 前記拡張型加算ブロックには、前記複数の部分積のうちその前記位取り部が最も小さな部分積が与えられる、請求項3記載のツリー回路。

【請求項5】 前記拡張型加算ブロックは、これに入力される前記第2要素のビット位置である特定ビット位置を含んだ上位において所定個数の拡張型加算器を、前記特定ビット位置よりも下位のビット位置において複数の通常型加算器を、それぞれ有し、前記拡張型加算器は、前記通常型加算器と比較して、自身よりも一つ上位のビットへと出力する上位伝搬出力の数が一つ多い、請求項4記載のツリー回路。

【請求項6】 前記拡張型加算ブロックは、前記拡張型加算器よりも上位においても加算器を有し、最も上位に配置された前記拡張型加算器に隣接した上位に配置された前記加算器は、前記上位伝搬出力の1つを、桁上げ入力以外の入力として受ける、請求項5記載のツリー回路。

【請求項7】 前記拡張型加算器は、その隣接する下位のビット位置から与えられる前記上位伝搬出力の他の入力の個数が4であり、その値が、全て“1”であるか否かに応じて、前記上位伝搬出力の1つが異なる値を採る、請求項5記載のツリー回路。

【請求項8】 複数の前記拡張型加算器同士において伝搬する前記上位伝搬出力は、疑似桁上げ出力として一对で生じ、かつ前記通常型加算器において生じる一对の桁上げに対して行われる所定の2つの演算の結果として表され得て、前記2つの演算の何れもが、前記一对の桁上げに対して可換である、請求項7記載のツリー回路。

【請求項9】 前記特定ビット位置に配置される前記拡張型加算器は、その隣接する下位のビット位置から的一对の桁上げ出力と、前記拡張型加算ブロックに入力する前記第2要素とを受け、その隣接する上位ビットに位置する前記拡張型加算器に前記疑似桁上げ出力を伝搬させる、請求項8記載のツリー回路。

【請求項10】 前記拡張型加算ブロックは、前記拡張型加算器よりも上位においても前記通常型加算器を有し、

10 前記拡張型加算器のうち最も上位に配置されるものは、その隣接する下位のビット位置から一对の前記疑似桁上げ出力を受け、その隣接する上位ビットに位置する前記通常型加算器に一对の桁上げ出力を与える、請求項9記載のツリー回路。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 この発明は、2値デジタル算術演算器に関し、特に、符号付き2の補数表現された乗数及び被乗数から符号付き2の補数表現された積を求める並列乗算回路に用いられるツリー回路に関するものである。

## 【0002】

【従来の技術】 今日の、マイクロプロセッサやDSPなどでは、乗算命令を高速に実行するために並列乗算器を搭載しているものが一般的である。並列乗算回路は、乗算の入力オペランドである乗数と被乗数から、複数の部分積を生成し、これら部分積を足し合わせることでより乗算結果である積出力を得るものである。よって、並列乗算回路の高速化手法としては、以下の2つの目的のためのアプローチがある。

【0003】 第1の目的は、生成する部分積の数を減らすことである。この目的を達成するためにはブースのアルゴリズム、特に2次のブースのアルゴリズムが代表的に用いられる。第2の目的は、生成された複数の部分積の加算を高速に実行することであり、このためには高速な加算回路を並列に動作させる回路方式が求められる。

【0004】 従来の技術による高速乗算回路を説明するため、まず32ビットの符号付き2の補数表現された乗数Yと、32ビットの符号付き2の補数表現された被乗数Xとを乗算し、64ビットの符号付き2の補数表現された積Zを求める回路（以下、32×32乗算器と略称する）を例に採る。

【0005】 乗数Yの各ビット毎に部分積を生成すると、部分積の数は32となり、これらを足し合わせる必要がある。しかし2次のブースのアルゴリズムを用いることにより、乗数Yにおいて隣り合う3ビットを組にして把握することにより、部分積の数を減らし、第1の目的を達成することができる。

【0006】 つまり、乗数Yを32ビットの符号付き2の補数で表現すると、 $y_i$  ( $i=0\sim31$ ) が0または

3

4

1を採用として、

【0007】

【数1】

$$Y = -y_{31} \cdot 2^{31} + \sum_{i=0}^{30} y_i \cdot 2^i$$

$$Z = X \cdot Y = \sum_{j=0}^{15} X \cdot (-2y_{2j+1} + y_{2j} + y_{2j-1}) \cdot 2^{2j} = \sum_{j=0}^{15} P_j$$

(但し、 $y_{-1} = 0$ )

【0008】となるので、積Zは、

【0009】

【数2】

【0010】と表すことができる。このようにして積Zを求めるのには16個の部分積 $P_j$  ( $j=0 \sim 15$ )を足し合わせるだけで足りる。

【0011】表1に、2次のブースのアルゴリズムの真

理値表を示す。

【0012】

【表1】

$y_{2j+1}$	$y_{2j}$	$y_{2j-1}$	$P_j$	$pp_j$	$pc_j$
0	0	0	0	0	0
0	0	1	$+X \cdot 2^j$	X	0
0	1	0	$+X \cdot 2^j$	X	0
0	1	1	$+X \cdot 2^{j+1}$	2X	0
1	0	0	$-X \cdot 2^{j+1}$	$\sim 2X$	1
1	0	1	$-X \cdot 2^j$	$\sim X$	1
1	1	0	$-X \cdot 2^j$	$\sim X$	1
1	1	1	0	0	0

【0013】乗数Yにおいて隣り合う3ビットのとりうる値の組み合わせは8通りある。よって、部分積 $P_j$ は、0、 $+X \cdot 2^j$ 、 $+X \cdot 2^{j+1}$ 、 $-X \cdot 2^j$ 、 $-X \cdot 2^{j+1}$ のいずれかの値を採用。2の補数表現された2値デジタル演算においては、あるデータを「2倍する」には1ビットだけ上位にそのデータ全体をシフトすればよく、「符号反転する」には、そのデータの各ビットを反転させ（これは部分積の第1要素 $pp_j$ の値の反転が対応する）、最下位ビットの桁に1を加算すればよい（これは部分積の第2要素 $pc_j$ が値“1”を採用することに対応する）。そして、部分積 $P_j$ は次のように表すことができる。

【0014】

【数3】

$$P_j = (pp_j + pc_j) \cdot 2^{2j}$$

【0015】よって、2次のブースのアルゴリズムによって生成された16の部分積 $P_0 \sim P_{15}$ を足し合わせるためには、 $j=0 \sim 15$ に対し、最下位桁が $2j$ ビット目である33ビット（つまり、ビット位置が $2j \sim 2j+32$ を占める。32ビットのデータが2倍される可能性があるため、32ビットよりも1ビット大きい位置が必要となる。）の部分積の第1要素 $pp_j$ 、及び、位置

が $2j$ ビット目である1ビットの部分積の第2要素 $pc_j$ を足し合わせる必要があることになる（つまり $2^{2j}$ は第1及び第2要素の位取りを示している）。

【0016】また、第2の目的を達成するための、即ち部分積の加算を高速に実行するための、高速な加算回路を並列に動作させる回路方式としては、キャリーセーブ法や、Wallace-Tree法等が代表的である。いずれも、複数（本例では16）の部分積を加算する際に、中間和の数を順次減少させつつ段階的に加算して最終的に2つの中間和（以下「最終中間和」という）を生成する。この最終中間和を得る迄に生じる桁上げ信号が後段の加算へと先送りされる。これにより、桁上げ信号が上位ビットへ伝搬することを並列化し、全体としてのクリティカルパス（回路動作を律速する経路）が短くなり、加算が高速となる。

【0017】生成された2つの最終中間和は、最終的に加算されて乗数と被乗数との積として得られる。この最終加算は2つの複数ビットの加算であり、例えばキャリールックアヘッド方式等を用いて高速化を図ることができる。かかる最終加算は周知の技術であるので詳細な説明は省略する。

【0018】以下では、第2の目的を達成するための、複数の部分積を順次減少させつつ段階的に加算して最終

的に2つの最終中間和を生成する技術について詳細な検討を行い、従来の技術の問題点を呈示する。

【0019】図18は、Wallace-Tree法を実現する従来の技術を示すブロック図であり、4入力2出力加算ブロック22a～22gが木(Tree)構造をとるように相互接続されている。そして、4入力2出力加算ブロック22gの出力が与えられる3入力2出力加算ブロック24aも備えられている。

【0020】図19乃至図21は三者相まって図18の詳細を示すブロック図である。図19は図20と仮想線 $Q_{19}$ 、 $Q_{19}$ において連続し、図20は図21と仮想線 $Q_{20}$ 、 $Q_{20}$ において連続する。各加算ブロックの横幅はそのブロックのビット幅に相当し、横方向の位置はビット位置を表している。

【0021】図22乃至図24は三者相まって4入力2出力加算ブロック22aの構成を例示するブロック図であり、図22は図23と仮想線 $Q_{22}$ 、 $Q_{22}$ において連続し、図23は図24と仮想線 $Q_{23}$ 、 $Q_{23}$ において連続する。4入力2出力加算ブロック22aは、1ビットに対応する4入力2出力加算器200の35個が直列接続されて構成されている。各ビット位置の4入力2出力加算器200の桁上げ出力 $C_0$ は、1つ上位に位置する4入力2出力加算器200の桁上げ入力 $C_i$ となる。1ピッ

A	B	C	S0	C0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

【0025】通常、排他的論理和ゲート(以降XOR)1段分の遅延は、他の論理積、論理和ゲートや複合ゲート1段分の遅延よりも大きく、2段分程度である。例えば、図25に示された構成における3入力2出力加算器400のクリティカルパスはXOR 2段である。

【0026】図26は4入力2出力加算器200の構成例を示すブロック図である。4入力2出力加算器200の1ビット分は、1ビットの3入力2出力加算器400を2つ用いて構成することができる。この場合にはそのクリティカルパスはXOR 4段である。

【0027】しかし、4入力2出力加算器200の構成を工夫することにより、更にここで要される遅延を小さくすることができる。表3は4入力2出力加算器200が満足すべき機能の一例を示す真理値表である。

【0028】

【表3】

トの4入力2出力加算器の機能として、桁上げ出力 $C_0$ を桁上げ入力 $C_i$ に依存しないようにすれば、これらを直列に接続して得られる4入力2出力加算ブロック内では、桁上げ出力 $C_0$ が、隣接するビットを越えて伝搬することはない。

【0022】

【発明が解決しようとする課題】しかし、図18乃至図21に示された従来の技術における部分積の加算では、1ビットの部分積の第2要素 $p_{c_{15}}$ を4入力2出力加算ブロック22gの下位出力 $s_{0,17}$ と上位出力 $c_{0,17}$ とに加算するだけのために、4段目に属する3入力2出力加算ブロック24aが必要とされる。そしてこの部分積の第2要素 $p_{c_{15}}$ が乗算器の高速化の妨げになっており、3入力2出力加算ブロック24aが集積化の妨げになっている。

【0023】具体的に、乗算器の速度を以下のようにして見積もってみる。図25は、それらが直列に接続されて3入力2出力加算ブロック24aを構成する3入力2出力加算器400の1ビット分の例を示す回路図であり、表2はその真理値表である。

【0024】

【表2】

A	B	C	D	S0	C0	C0
0	0	0	0	$\sim C_i$	0	0
0	0	0	1	$\sim C_i$	0	$C_i$
0	0	1	0	$\sim C_i$	0	$C_i$
0	0	1	1	$\sim C_i$	0	1
0	1	0	0	$\sim C_i$	0	$C_i$
0	1	0	1	$\sim C_i$	1	0
0	1	1	0	$\sim C_i$	1	0
0	1	1	1	$\sim C_i$	1	$C_i$
1	0	0	0	$\sim C_i$	0	$C_i$
1	0	0	1	$\sim C_i$	1	0
1	0	1	0	$\sim C_i$	1	0
1	0	1	1	$\sim C_i$	1	$C_i$
1	1	0	0	$\sim C_i$	0	1
1	1	0	1	$\sim C_i$	1	$C_i$
1	1	1	0	$\sim C_i$	1	$C_i$
1	1	1	1	$\sim C_i$	1	1

【0029】図27は表3を満たす回路の一例を示す回路図である。クリティカルパスは4入力2出力加算器200の出力S0を得る経路である。出力S0は4つの入

力A, B, C, Dと桁上げ入力C<sub>i</sub>の5つの信号の排他的論理和であるが、図27から分かるように、入力A, Bの排他的論理和と、入力C, Dの排他的論理和とは並列して処理できるので、結局クリティカルパスはXOR3段である。そこで簡単のため、以降では3入力2出力加算器400の遅延はXOR2段、4入力2出力加算器200の遅延はXOR3段として説明を進める。

【0030】既述のように、桁上げ出力C<sub>0</sub>は隣接するビットを越えて伝搬しないので、加算器200, 400の遅延時間はそのままそれぞれ加算ブロック22a~22g, 24aの遅延時間を決定する。

【0031】乗算器の入力である乗数Yと被乗数Xとが同時に入力されると仮定すれば、2次のブースのアルゴリズムによって生成される部分積は、その第1要素p<sub>p0</sub>~p<sub>p15</sub>及び第2要素p<sub>c0</sub>~p<sub>c15</sub>は同時にその値が確定する。

【0032】よって、図18乃至図21に示された加算ブロックのうち、Tree回路の1段目に属するもの(4入力2出力加算ブロック22a~22d)、2段目に属するもの(4入力2出力加算ブロック22e, 22f)、3段目に属するもの(4入力2出力加算ブロック22g)、4段目に属するもの(3入力2出力加算ブロック24a)の順番に加算処理が行われる。従って、部分積の第1要素p<sub>p0</sub>~p<sub>p15</sub>及び第2要素p<sub>c0</sub>~p<sub>c15</sub>が確定してから、2つの最終中間和となる、3入力2出力加算ブロック24aの下位出力s<sub>0,0</sub>及び上位出力c<sub>0,0</sub>が確定するまでの遅延時間は、4入力2出力加算ブロック3段と3入力2出力加算ブロック1段を経由するので、XOR11(=3×3+2)段分となる。

【0033】図18では、16個の部分積を足し合わせて2つの最終中間和を生成するTree回路として4入力2出力加算ブロックを基本とした構成を示したが、Treeの最終段以外にも3入力2出力加算ブロックを用いて構成することもできる。

【0034】図28は、3入力2出力加算ブロックの占める割合を多くしたTree回路の構造を示すブロック図である。図29乃至図32は四者相まって図28の詳細を示すブロック図である。図29は図30と仮想線Q<sub>2,0</sub>, Q<sub>2,1</sub>において連続し、図30は図31と仮想線Q<sub>3,0</sub>, Q<sub>3,1</sub>において連続し、図31は図32と仮想線Q<sub>3,1</sub>, Q<sub>3,1</sub>において連続する。図19乃至図21と同様に、各加算ブロックの横幅はそのブロックのビット幅に相当し、横方向の位置はビット位置を表している。

【0035】4入力2出力加算ブロック32a~32dの出力を3入力2出力加算ブロック34a~34cが受け、3入力2出力加算ブロック34a~34cの出力を3入力2出力加算ブロック34d~34eが受け、3入力2出力加算ブロック34d~34eの出力を4入力2出力加算ブロック32eが受ける。そして4入力2出力加算ブロック32eが2つの最終中間和として下位出力

s<sub>0,0</sub>、上位出力c<sub>0,0</sub>を出力する。

【0036】図18を用いて示された場合と異なり、第2要素p<sub>c<sub>i</sub></sub>はjの順に並べて集められ、p<sub>p<sub>c</sub></sub>として4入力2出力加算ブロック32aに与えられる。つまり【0037】

【数4】

$$p_{pc} = \sum_{j=0}^{15} p_{c_j} \cdot 2^{2j}$$

10 【0038】である。

【0039】図28乃至図32における加算ブロックのうち、Tree回路の1段目に属するもの(4入力2出力加算ブロック32a~32d)、2段目に属するもの(3入力2出力加算ブロック34a~34c)、3段目に属するもの(3入力2出力加算ブロック34d~34e)、4段目に属するもの(4入力2出力加算ブロック32e)の順番に加算処理が行われる。従って、部分積の第1要素p<sub>p0</sub>~p<sub>p15</sub>及び第2要素p<sub>c0</sub>~p<sub>c15</sub>が確定してから、2つの最終中間和となる、4入力2出力加算ブロック32eの下位出力s<sub>0,0</sub>及び上位出力c<sub>0,0</sub>が確定するまでの遅延時間は、4入力2出力加算ブロック2段と3入力2出力加算ブロック2段を経由するので、XOR10(=3×2+2×2)段分となる。図18乃至図21で示された構成と比較して、遅延時間が改善されている。

【0040】しかし、加算ブロックの数は2個増加しており、回路規模が大きくなってしまいうという欠点がある。これは3入力2出力加算ブロックが、4入力2出力加算ブロックよりも遅延時間が短いものの、並列処理できる入力数が1つ少ないことに起因する。

【0041】図33は、いずれも24ビットの符号付き2の補数表現された乗数及び被乗数を乗算し、48ビットの符号付き2の補数表現された積を求める回路における最終中間和を生成するTree回路を例示するブロック図である。2次のブースのアルゴリズムを用いた場合に得られる部分積の数は12個であり、これらを順次減少させつつ段階的に加算して最終的に2つの最終中間和を生成する。

【0042】4入力2出力加算ブロック42a~42e及び3入力2出力加算ブロック44aでTree回路が構成されている。Tree回路の1段目に属するもの(4入力2出力加算ブロック42a~42c)、2段目に属するもの(4入力2出力加算ブロック42d、3入力2出力加算ブロック44a)、3段目に属するもの(4入力2出力加算ブロック42e)の順番に加算処理が行われる。

p<sub>p<sub>c</sub></sub>は

【0043】

【数5】

$$p p c = \sum_{j=0}^{11} p c_j \cdot 2^{2j}$$

【0044】として表される。

【0045】4入力2出力加算ブロック3段を経由する方(42a(または42b), 42d, 42eを通る経路)が、4入力2出力加算ブロック2段と3入力2出力加算ブロック1段を経由する方(42c, 44a, 42eを通る経路)よりも遅延時間が長い。従って、部分積の第1要素 $p p_0 \sim p p_{11}$ 及び第2要素 $p c_0 \sim p c_{11}$ が確定してから、2つの最終中間和となる4入力2出力加算ブロック42eの下位出力 $s o_{16}$ 及び上位出力 $c o_{16}$ が確定するまでの遅延時間は、4入力2出力加算ブロックの3段分に相当し、XOR 9段分となる。

【0046】ここで、上述のように4入力2出力加算ブロック42eに入力するデータの確定時刻は等しくない。即ち3入力2出力加算ブロック44aの下位出力 $s o_{16}$ と上位出力 $c o_{16}$ が、4入力2出力加算ブロック42dの下位出力 $s o_{16}$ と上位出力 $c o_{16}$ よりもXOR 1段分早く確定している。更に、3入力2出力加算ブロック44aに入力するデータについて言えば、部分積の第1要素 $p p_{11}$ が、4入力2出力加算ブロック42cの下位出力 $s o_{16}$ と上位出力 $c o_{16}$ よりもXOR 3段分早く確定している。

【0047】このように、従来の技術において、乗算の入力データのビット幅によってはTree回路の回路動作の並列度が低い場合があった。換言すればTree回路を構成する回路ブロックの入力データが確定するタイミングが不揃いであるため、乗算器の高速化の妨げとなるという問題点もあった。

【0048】本発明は、上記のような問題点を解消するためになされたもので、2次のプースのアルゴリズムを用いた並列乗算回路の回路動作の並列度を高め、回路規模を著しく増大させることなく乗算器を高速化することを目的とする。

【0049】

【課題を解決するための手段】この発明のうち請求項1にかかるものは、プースのアルゴリズムに従って生成された複数の部分積に基づいて累進的な加算を行うことにより、個数が逓減する中間和を生成しつつ、一対の最終中間和を出力するツリー回路である。そして複数のビット幅のデータの複数の加算して一対の前記中間和を出力する通常型加算ブロックと、複数のビット幅のデータの複数のと、1ビット幅のデータとを加算して一対の前記中間和を出力する拡張型加算ブロックとを備える。

【0050】この発明のうち請求項2にかかるものは、請求項1記載のツリー回路であって、各々の前記部分積は、複数ビットの第1要素と1ビットの第2要素との和に対して自身の最下位ビット位置を示す位取り部が乗じられたものとして表される。前記拡張型加算ブロック

は、前記複数の部分積を入力し、自身に入力される前記複数の部分積以外の前記複数の部分積に属する第2要素の一つを更に入力する。

【0051】この発明のうち請求項3にかかるものは、請求項2記載のツリー回路であって、前記拡張型加算ブロックに入力される前記第2要素は、前記複数の部分積のうち前記位取り部が最も大きな部分積に属する。

【0052】この発明のうち請求項4にかかるものは、請求項3記載のツリー回路であって、前記拡張型加算ブロックには、前記複数の部分積のうちその前記位取り部が最も小さな部分積が与えられる。

【0053】この発明のうち請求項5にかかるものは、請求項4記載のツリー回路であって、前記拡張型加算ブロックは、これに入力される前記第2要素のビット位置である特定ビット位置を含んだ上位において所定個数の拡張型加算器を、前記特定ビット位置よりも下位のビット位置において複数の通常型加算器を、それぞれ有する。前記拡張型加算器は、前記通常型加算器と比較して、自身よりも一つ上位のビットへと出力する上位伝搬出力の数が一つ多い。

【0054】この発明のうち請求項6にかかるものは、請求項5記載のツリー回路であって、前記拡張型加算ブロックは、前記拡張型加算器よりも上位においても加算器を有する。最も上位に配置された前記拡張型加算器に隣接した上位に配置された前記加算器は、前記上位伝搬出力の1つを、桁上げ入力以外の入力として受ける。

【0055】この発明のうち請求項7にかかるものは、請求項5記載のツリー回路であって、前記拡張型加算器は、その隣接する下位のビット位置から与えられる前記上位伝搬出力の他の入力の個数が4であり、その値が全て“1”であるか否かに応じて、前記上位伝搬出力の1つが異なる値を採る。

【0056】この発明のうち請求項8にかかるものは、請求項7記載のツリー回路であって、複数の前記拡張型加算器同士において伝搬する前記上位伝搬出力は、疑似桁上げ出力として一対で生じ、かつ前記通常型加算器において生じる一対の桁上げに対して行われる所定の2つの演算の結果として表され得る。そして前記2つの演算の何れもが、前記一対の桁上げに対して可換である。

【0057】この発明のうち請求項9にかかるものは、請求項8記載のツリー回路であって、前記特定ビット位置に配置される前記拡張型加算器は、その隣接する下位のビット位置からの一対の桁上げ出力と、前記拡張型加算ブロックに入力する前記第2要素とを受け、その隣接する上位ビットに位置する前記拡張型加算器に前記疑似桁上げ出力を伝搬させる。

【0058】この発明のうち請求項10にかかるものは、請求項9記載のツリー回路であって、前記拡張型加算ブロックは、前記拡張型加算器よりも上位においても前記通常型加算器を有する。前記拡張型加算器のうち最

も上位に配置されるものは、その隣接する下位のビット位置から一対の前記疑似桁上げ出力を受け、その隣接する上位ビットに位置する前記通常型加算器に一対の桁上げ出力を与える。

【0059】

【発明の実施の形態】

実施の形態1. 図1は本発明の実施の形態1にかかる乗算器の構成の一部を示すブロック図である。乗数、被乗数は共に32ビットで符号付き2の補数表現され、2次のブースのアルゴリズムを用いて16個の部分積 $P_0 \sim P_{15}$ が得られる。ここではこれらの部分積を生成する部分は示さず、部分積を加算する際に、中間和の数を順次段階的に減少させ、最終的に2つの最終中間和を生成するTree回路の概略を示している。従来の技術において説明されたように数3に基づいて、部分積 $P_i$ は33ビット幅の第1要素 $p_{pi}$ と、1ビット幅の第2要素 $p_{ci}$ と、これらの最下位ビットの位置を示す $2^j$ によって決定される。

【0060】実施の形態1において示されたTree回路は、複数ビット幅の4つの入力データと1ビットの入力データとを同時に加算する回路ブロック（以降「拡張型4入力2出力加算ブロック」と称する）1aと、4入力2出力加算ブロック2a～2fとから構成されている。

【0061】従来の技術において3入力2出力加算ブロック24aに与えられていた部分積の第2要素 $p_{ci15}$ は、本実施の形態においては拡張型4入力2出力加算ブロック1aに与えられる。このため、本実施の形態では3入力2出力加算ブロック24aは必要ない。

【0062】拡張型4入力2出力加算ブロック1aは、部分積の第2要素 $p_{ci15}$ と、部分積の第1要素 $p_{p0} \sim p_{p15}$ を入力し、上位出力 $c_{o1}$ 及び下位出力 $s_{o1}$ を出力する。4入力2出力加算ブロック2aは部分積の第1要素 $p_{p4} \sim p_{p7}$ を入力し、中間和として上位出力 $c_{o2}$ 及び下位出力 $s_{o2}$ を出力する。4入力2出力加算ブロック2bは部分積の第1要素 $p_{p8} \sim p_{p11}$ を入力し、中間和として上位出力 $c_{o3}$ 及び下位出力 $s_{o3}$ を出力する。4入力2出力加算ブロック2cは部分積の第1要素 $p_{p12} \sim p_{p15}$ を入力し、中間和として上位出力 $c_{o4}$ 及び下位出力 $s_{o4}$ を出力する。4入力2出力加算ブロック2dは上位出力 $c_{o1}$ 、 $c_{o2}$ 及び下位出力 $s_{o1}$ 、 $s_{o2}$ を入力し、中間和として上位出力 $c_{o5}$ 及び下位出力 $s_{o5}$ を出力する。4入力2出力加算ブロック2eは上位出力 $c_{o3}$ 、 $c_{o4}$ 及び下位出力 $s_{o3}$ 、 $s_{o4}$ を入力し、中間和として上位出力 $c_{o6}$ 及び下位出力 $s_{o6}$ を出力する。4入力2出力加算ブロック2fは上位出力 $c_{o5}$ 、 $c_{o6}$ 及び下位出力 $s_{o5}$ 、 $s_{o6}$ を入力し、最終中間和として上位出力 $c_{o7}$ 及び下位出力 $s_{o7}$ を出力する。下位出力 $s_{o7}$ と上位出力 $c_{o7}$ とは、図示しない最終加算ブロックによって最終加算されて乗算結果が得られる。但し、以上の関係は概

略的なものであり、詳細には図2乃至図4を用いて後述される。

【0063】図2乃至図4は三者相まって図1の詳細を示すブロック図である。図2と図3とは仮想線 $Q_2$ 、 $Q_2$ において連続し、図3と図4とは仮想線 $Q_3$ 、 $Q_3$ において連続する。各加算ブロックの横幅はそのブロックのビット幅に相当し、横方向の位置はビット位置を表している。

【0064】部分積 $P_0$ の第1要素 $p_{p0} < 32 : 0 >$ （ $< u : v >$ はその前に配置された符号によって表されるデータが、乗算結果の最下位たる第0ビット目から数えて、 $v$ ビット目～ $u$ ビット目の位置を占めることを示す。）のうち第2ビット目～第32ビット目と、部分積 $P_1$ の第1要素 $p_{p1} < 34 : 2 >$ の全てと、部分積 $P_1$ の第2要素 $p_{c1}$ と、部分積 $P_2$ の第1要素 $p_{p2} < 36 : 4 >$ の全てと、部分積 $P_2$ の第2要素 $p_{c2}$ と、部分積 $P_3$ の第1要素 $p_{p3} < 38 : 6 >$ のうち第6ビット目～第36ビット目と、部分積 $P_{15}$ の第2要素 $p_{c15}$ とが、ビット位置を整合させて拡張型4入力2出力加算ブロック1aに与えられる。

【0065】部分積 $P_1$ の第2要素 $p_{c1}$ と、部分積 $P_2$ の第2要素 $p_{c2}$ とは部分積 $P_3$ の第1要素 $p_{p3}$ の疑似的な下位として扱われる。

【0066】部分積 $P_0$ の第2要素 $p_{c0}$ は、これが占めるビット位置（第0ビット目）には他のデータが位置しないので、図示しない最終加算ブロックへと送られる。

【0067】部分積 $P_3$ の第1要素 $p_{p3}$ のうちの第37ビット目及び第38ビット目は、これが占めるビット位置には拡張型4入力2出力加算ブロック1aが対応しないので、拡張型4入力2出力加算ブロック1aの下位出力 $s_{o1} < 36 : 2 >$ の疑似的な上位として扱われ、4入力2出力加算ブロック2dへと送られる。

【0068】第2要素 $p_{c3}$ は、これが占めるビット位置（第6ビット目）に対応するデータが既に4つ存在するので、拡張型4入力2出力加算ブロック1aにおいて加算されはしない。これは4入力2出力加算ブロック2dへと送られる。

【0069】部分積 $P_0$ の第1要素 $p_{p0}$ の第0ビット目及び第1ビット目はこれが占めるビット位置には拡張型4入力2出力加算ブロック1aが対応しないので、拡張型4入力2出力加算ブロック1aの下位出力 $s_{o1}$ の疑似的な下位として扱われる。

【0070】第2要素 $p_{c1}$ 、 $p_{c2}$ が配置されるビット位置においては、それぞれ第1要素 $p_{p2}$ 、 $p_{p3}$ が配置されないで、拡張型4入力2出力加算ブロック1aにおいて、この位置には従来の4入力2出力加算器200を用いることができる。しかし、第2要素 $p_{c15}$ が配置されるビット位置、つまり30ビット目において



は、4つの第1要素 $p p_0 \sim p p_3$ も配置される。このため、少なくともこの位置における拡張型4入力2出力加算ブロック1aは、6入力を扱う加算器を備えていなければならない。つまり4つの第1要素 $p p_0 < 30 >$ ,  $p p_1 < 30 >$ ,  $p p_2 < 30 >$ ,  $p p_3 < 30 >$  ( $< w >$ はビット位置を示す)と、第2要素 $p c_{15}$ と、第29ビット目に配置された4入力2出力加算器200の桁上げ出力 $C o$ との6つである。

【0071】しかも、この第30ビット目に配置される加算器(以下「拡張型4入力2出力加算器」と称する)の出力は桁上げ出力が2つ必要となる。1ビットのデータが6個入力するのであるから、その加算結果の最大値は10進数で6である。桁上げ出力を隣接するビットにのみ伝搬させるためには入力されるデータのビット位置に対して更に $2^1$ の重みを有する桁上げのみで処理を行い、 $2^2$ の重みを有する桁上げを用いて処理することは許されないためである。勿論、加算結果が10進数でいう奇数になる場合もあるので、入力されるデータのビット位置に対して $2^0$ の重みを有する(即ち当該ビット位置に対応する)出力も必要である。よって、拡張型4入力2出力加算器は、 $2^0$ の重みを有する下位出力 $S o$ 、 $2^1$ の重みを有する上位出力 $C o$ (これらはそれぞれ4入力2出力加算ブロックの下位出力 $s o$ 、上位出力 $c o$ の1ビット分に相当する)を出力し、更にいずれも $2^1$ の重みを有する第1の桁上げ出力 $C o 1$ 、第2の桁上げ出力 $C o 2$ が一つ上位に位置する加算器に送られる。

【0072】このため、第1の原則として第2要素 $p c_{15}$ が配置されるビット位置よりも上位に配置される加算器は、4つの第1要素 $p p_0 \sim p p_3$ と、下位において隣接する拡張型4入力2出力加算器から与えられる第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ とを入力する必要がある、やはり拡張型4入力2出力加算器でなければならない。

【0073】但し、第1要素 $p p_0$ の最上位ビットは第32ビット目に位置しているので、第33ビット目以上の拡張型4入力2出力加算ブロック1aにおいては、入力が一つ不要となっている。従って、第2の原則として第33ビット目以上では、下位からの桁上げ出力が2つ存在しても、(通常の)4入力2出力加算器を用いて拡張型4入力2出力加算ブロック1aを構成することができる。

【0074】つまり第1及び第2の原則に従えば、最もビット位置の高い部分積 $P_{15}$ の第2要素 $p c_{15}$ が配置されるビット位置を含んでこれよりも大きく、最もビット位置の低い部分積 $P_0$ の第1要素 $p p_0$ が占める最上位のビット位置以下のビット位置に対して、従来の4入力2出力加算ブロックにおける4入力2出力加算器200を拡張型4入力2出力加算器に置換すれば良いことになる。

【0075】但し、第1及び第2の原則以外に後述する

ように「1加算法」に因る補正があり、拡張型4入力2出力加算器は更に上位のビット位置においても必要となってくる。以下、拡張型4入力2出力加算ブロックの構成について詳細に説明すると共に上記補正について簡単に説明する。

【0076】図5乃至図7は三者相まって拡張型4入力2出力加算ブロック1aの構成を例示するブロック図である。図5及び図6は仮想線 $Q_5$ 、 $Q_6$ において連続し、図6及び図7は仮想線 $Q_6$ 、 $Q_7$ において連続する。

10 【0077】拡張型4入力2出力加算ブロック1aは35ビットの加算を並列して行うため、1ビットの拡張型4入力2出力加算器100が第30ビット目～第34ビット目において5個配置され、1ビットの4入力2出力加算器200が第2ビット目～第29ビット目において28個、第35ビット目及び第36ビット目において2個配置されている。

20 【0078】第0ビット目の4入力2出力加算器200の桁上げ入力 $C i$ には下位からの桁上げがないので“0”が入力されている。そして順次、4入力2出力加算器200の桁上げ出力 $C i$ はその上位において隣接する4入力2出力加算器200の桁上げ入力 $C i$ となっている。

30 【0079】但し、第29ビット目の4入力2出力加算器200の桁上げ出力 $C o$ は、その上位において隣接する拡張型4入力2出力加算器100へ、その第2桁上げ入力 $C i 2$ として与えられる。そして第30ビット目～第33ビット目の拡張型4入力2出力加算器100の第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ は、それぞれ第31ビット目～第34ビット目の拡張型4入力2出力加算器100の第1及び第2の桁上げ入力 $C i 1$ 、 $C i 2$ として与えられる。

40 【0080】第30ビット目の拡張型4入力2出力加算器100の第1の桁上げ入力 $C i 1$ には部分積 $P_{15}$ の第2要素 $p c_{15}$ が入力されている。第1の桁上げ入力 $C i 1$ も、この第30ビット目というビット位置の重みを持つという点では4つの第1要素 $p p_0 < 30 >$ ,  $p p_1 < 30 >$ ,  $p p_2 < 30 >$ ,  $p p_3 < 30 >$ と同格であり、上記第1の原則に従ったものである。勿論同じ理由で第30ビット目の拡張型4入力2出力加算器100の第2の桁上げ入力 $C i 2$ に部分積 $P_{15}$ の第2要素 $p c_{15}$ を入力し、第1の桁上げ入力 $C i 1$ に第29ビット目の4入力2出力加算器200の桁上げ出力 $C o$ を与えても良い。

50 【0081】第34ビット目の拡張型4入力2出力加算器100の第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ は、それぞれ第35ビット目の4入力2出力加算器200の4入力の内の一つ(図において“D”と示された入力)と桁上げ入力 $C i$ として与えられる。第35ビット目の4入力2出力加算器200の桁上げ出力 $C o$ は第36ビット目の4入力2出力加算器200の桁上げ入力 $C$



iとして与えられる。

【0082】4つの第1要素 $p p_0 \sim p p_3$ はそれぞれビット位置を整合させつつ、対応するビット位置の4入力2出力加算器100、200の4入力A～Dに与えられる。そしてビット毎に4入力2出力加算器100、200はそれぞれ上位出力C0と下位出力S0とを出力し、これらはそれぞれ拡張型4入力2出力加算ブロック1aの上位出力 $c o 1$ と下位出力 $s o 1$ の各ビットに対応している。

【0083】但し、第1要素 $p p_2$ には第2及び第3ビット目に配置されるべきデータを有さず、第1要素 $p p_3$ には第2乃至及び第5ビット目に配置されるべきデータを有さない。その一方で第2要素 $p c_1$ 、 $p c_2$ がそれぞれ第2ビット目、第4ビット目に配置される。従って、第2ビット目の4入力2出力加算器200のA入力には第2要素 $p c_1$ が、B入力には“0”が、第3ビット目の4入力2出力加算器200のA入力及びB入力には“0”が、第4ビット目の4入力2出力加算器200のA入力には第2要素 $p c_2$ が、第5ビット目の4入力2出力加算器200のA入力には“0”が、それぞれ与えられることになる。

【0084】第1及び第2の原則に従うのであれば、拡張型4入力2出力加算器100は第30ビット目乃至第32ビット目の3個で済むはずであり、第33ビット目及び第34ビット目においては（通常の）4入力2出力加算器200が配置されるはずである。しかし、図5に示されているように、実は第32ビット目の拡張型4入力2出力加算器100のD入力には第1要素 $p p_0 < 3 2 >$ ではなくその反転した値が与えられ、第33ビット目の拡張型4入力2出力加算器100のD入力には“0”ではなく第1要素 $p p_0 < 3 2 >$ の反転した値が与えられ、第34ビット目の拡張型4入力2出力加算器100のD入力には“0”ではなく第1要素 $p p_0 < 3 2 >$ が与えられ、そして第35ビット目の4入力2出力加算器200のC入力には“0”ではなく“1”が与えられている。

【0085】このような、2の補数表示を行った符号付きの数の演算における被加算数の変更は、ここでは詳細を示さないが、符号ビット拡張を簡略化するための「1加算法」と呼ばれる周知の技法である。通常かかる技法が用いられるため、本実施の形態においても拡張型4入力2出力加算器100は第1及び第2の原則に必要な個数よりも所定の個数だけ上位側にも必要となる（第1及び第2の原則に対する補正）。この所定の個数は、部分積を生成するためのブースのアルゴリズムが何次であるかに依存し、ここでは2個である。

【0086】第35ビット目において与えられるべき第1要素は $p p_2 < 3 5 >$ 、 $p p_3 < 3 5 >$ の2つであり、「1加算法」を施すことに因って入力するのが必要な“1”と、第34ビット目の第1の桁上げ出力 $C o 1$

を含めて4つの入力を加算すればよいので、（通常の）4入力2出力加算器200を用いることができる。勿論第36ビット目においては第35ビット目の4入力2出力加算器200からの桁上げ出力 $C o$ を桁上げ入力 $C i$ として入力し、第1要素 $p p_0$ 、 $p p_1$ はこのビット位置においては存在しないので、やはり（通常の）4入力2出力加算器200を用いることができる。

【0087】勿論、第35ビット目の4入力2出力加算器200において、その桁上げ入力 $C i$ と、C入力と、D入力とは互いに等しい重みを有しているので互いに入れ換えることができる。また、拡張型4入力2出力加算器100の第1及び第2の桁上げ入力 $C i 1$ 、 $C i 2$ を互いに入れ換えることもできる。

【0088】さて、上記の拡張型4入力2出力加算器100は以下のようにして実現することができる。拡張型4入力2出力加算器100は実は6つの1ビットデータを入力し、その位置するビットに対応した下位出力 $S O 1$ つと、その位置するビットよりも1つ上位のビットに対応した3つの出力、即ち第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ 並びに上位出力 $C O$ を出力する。つまり、

【0089】

【数6】

$$A+B+C+D+C i 1+C i 2=2(C o 1+C o 2+C O)+S O$$

【0090】が成立する。

【0091】下位側で隣接する拡張型4入力2出力加算器100の第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ の値の和は10進法でいえば高々“2”であり、その与える影響は当該拡張型4入力2出力加算器100自身が位置するビットにおいて出力する上位出力 $C O$ と、下位出力 $S O$ のみに限定することができる。換言すれば、第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ は4つの入力A～Dのみに依存させることができる。これによって自身が位置するビット位置よりも2以上上位のビット位置へは桁上げが伝搬することがない。

【0092】また、第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ は4つの入力A～Dにおける“1”の数によってのみ決め、桁上げ入力 $C i 1$ 、 $C i 2$ の値は上位出力 $C O$ に反映させることができる。4つの入力A～Dにおける“1”の数が1つ以下であれば桁上げは生じないので、第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ はいずれも“0”とする。また“1”の数が2つまたは3つであれば第1の桁上げ出力 $C o 1$ のみを“1”とし、第2の桁上げ出力 $C o 2$ を“0”とする。そして“1”の数が4つであれば、第1及び第2の桁上げ出力 $C o 1$ 、 $C o 2$ をいずれも“1”とする。

【0093】一方、下位出力 $S O$ は当該拡張型4入力2出力加算器100自身の出力が10進法でいう偶数か奇数かを決定するのであるから、4つの入力A～Dと第1および第2の桁上げ入力 $C i 1$ 、 $C i 2$ における“1”の個数が偶数か奇数かで値が定まる。

【0094】また、上位出力COは第1及び第2の桁上げ入力Ci1, Ci2が共に“1”の場合には“1”である。また4つの入力A～Dにおける“1”の数が奇数のときには、第1および第2の桁上げ入力Ci1, Ci2のいずれかが“1”の場合にも“1”である。それ

A	B	C	D	S0	Co1	Co2	CO
0	0	0	0	Ci1·Ci2	0	0	Ci1 & Ci2
0	0	0	1	~(Ci1·Ci2)	0	0	Ci1 & Ci2
0	0	1	0	~(Ci1·Ci2)	0	0	Ci1 & Ci2
0	0	1	1	Ci1·Ci2	1	0	Ci1 & Ci2
0	1	0	0	~(Ci1·Ci2)	0	0	Ci1 & Ci2
0	1	0	1	Ci1·Ci2	1	0	Ci1 & Ci2
0	1	1	0	Ci1·Ci2	1	0	Ci1 & Ci2
0	1	1	1	~(Ci1·Ci2)	1	0	Ci1 & Ci2
1	0	0	0	~(Ci1·Ci2)	0	0	Ci1 & Ci2
1	0	0	1	Ci1·Ci2	1	0	Ci1 & Ci2
1	0	1	0	Ci1·Ci2	1	0	Ci1 & Ci2
1	0	1	1	~(Ci1·Ci2)	1	0	Ci1 & Ci2
1	1	0	0	Ci1·Ci2	1	0	Ci1 & Ci2
1	1	0	1	~(Ci1·Ci2)	1	0	Ci1 & Ci2
1	1	1	0	~(Ci1·Ci2)	1	0	Ci1 & Ci2
1	1	1	1	Ci1·Ci2	1	1	Ci1 & Ci2

以外は“0”となる。

【0095】これらを纏めると、

【0096】

【表4】

【0097】が得られる。つまり表4は拡張型4入力2出力加算器100が満足すべき入出力関係を例示する真理値表の第1の例である。表4の真理値表をブール式で

20 表現すると、

【0098】

【数7】

$$Co1 = (A \cdot B) \cdot (C \cdot D) \cdot (A \cdot B \cdot C \cdot D)$$

$$Co2 = A \cdot B \cdot C \cdot D$$

$$S0 = A \cdot B \cdot C \cdot D \cdot Ci1 \cdot Ci2$$

$$CO = \sim(A \cdot B \cdot C \cdot D) \cdot (Ci1 \cdot Ci2) \cdot (A \cdot B \cdot C \cdot D) \cdot (Ci1 \cdot Ci2)$$

記号“~”は論理反転、“·”は論理和、“&”は論理積、“...”は排他的論理和を表す

【0099】となる。

【0100】図8は数7に基づいて構成された、拡張型4入力2出力加算器100の構成の第1例を示す回路図である。NANDゲートG1には入力A～Dが与えられ、その出力はインバータG2に与えられ、インバータG2は第2の桁上げ出力Co2を出力する。

【0101】ORゲートG3には入力A, Bが与えられ、ORゲートG4には入力C, Dが与えられ、ORゲートG3, G4の出力はNANDゲートG5に与えられる。ゲートG3～G5は複合ゲートとして構成することができる。

【0102】ANDゲートG7には入力A, Bが与えられ、ANDゲートG8には入力C, Dが与えられ、ANDゲートG7, G8の出力はNORゲートG9に与えられる。ゲートG7～G9は複合ゲートとして構成することができる。

【0103】NANDゲートG5の出力とNORゲートG9の出力とはNANDゲートG6に与えられ、NANDゲートG6は第1の桁上げ出力Co1を出力する。

【0104】XORゲートG17には入力A, Bが与えられ、XORゲートG18には入力C, Dが与えられる。XORゲートG17, G18の出力はXORゲート

G19に与えられる。第1及び第2の桁上げ入力Ci1, Ci2はXORゲートG20に与えられ、XORゲートG19, G20の出力はXORゲートG21に与えられる。XORゲートG21は下位出力S0を出力する。

【0105】第1及び第2の桁上げ入力Ci1, Ci2はNANDゲートG10に与えられる。第1及び第2の桁上げ入力Ci1, Ci2はNORゲートG11にも与えられ、その出力はインバータG12に与えられる。NANDゲートG10の出力はXORゲートG19の出力と共にNORゲートG13に与えられる。インバータG12の出力はXORゲートG19の出力と共にANDゲートG14に与えられる。NORゲートG13の出力はANDゲートG14の出力と共にNORゲートG15に与えられ、NORゲートG15の出力はインバータG16に与えられる。ゲートG14, G15は複合ゲートとして構成することができる。インバータG16は上位出力COを出力する。

【0106】第1及び第2の桁上げ出力Co1, Co2は、互いにその重みが等しいので、採る値を互いに入れ換えても良い。“1”の数が2つであれば第1及び第2の桁上げ出力Co1, Co2をそれぞれ“1”, “0”

とし、3つであれば第1及び第2の桁上げ出力C<sub>0</sub>1、  
C<sub>0</sub>2をそれぞれ“0”，“1”とすることができる。  
【0107】かかる変更を表4に施して、

A	B	C	D	SO	Co1	Co2	CO
0	0	0	0	C <sub>11</sub> ·C <sub>12</sub>	0	0	C <sub>11</sub> & C <sub>12</sub>
0	0	0	1	~(C <sub>11</sub> ·C <sub>12</sub> )	0	0	C <sub>11</sub> C <sub>12</sub>
0	0	1	0	~(C <sub>11</sub> ·C <sub>12</sub> )	0	0	C <sub>11</sub> C <sub>12</sub>
0	0	1	1	C <sub>11</sub> ·C <sub>12</sub>	1	0	C <sub>11</sub> & C <sub>12</sub>
0	1	0	0	~(C <sub>11</sub> ·C <sub>12</sub> )	0	0	C <sub>11</sub> C <sub>12</sub>
0	1	0	1	C <sub>11</sub> ·C <sub>12</sub>	1	0	C <sub>11</sub> & C <sub>12</sub>
0	1	1	0	C <sub>11</sub> ·C <sub>12</sub>	1	0	C <sub>11</sub> & C <sub>12</sub>
0	1	1	1	~(C <sub>11</sub> ·C <sub>12</sub> )	0	1	C <sub>11</sub> + C <sub>12</sub>
1	0	0	0	~(C <sub>11</sub> ·C <sub>12</sub> )	0	0	C <sub>11</sub> + C <sub>12</sub>
1	0	0	1	C <sub>11</sub> ·C <sub>12</sub>	1	0	C <sub>11</sub> & C <sub>12</sub>
1	0	1	0	C <sub>11</sub> ·C <sub>12</sub>	1	0	C <sub>11</sub> & C <sub>12</sub>
1	0	1	1	~(C <sub>11</sub> ·C <sub>12</sub> )	0	1	C <sub>11</sub> + C <sub>12</sub>
1	1	0	0	C <sub>11</sub> ·C <sub>12</sub>	1	0	C <sub>11</sub> & C <sub>12</sub>
1	1	0	1	~(C <sub>11</sub> ·C <sub>12</sub> )	0	1	C <sub>11</sub> + C <sub>12</sub>
1	1	1	0	~(C <sub>11</sub> ·C <sub>12</sub> )	0	1	C <sub>11</sub> + C <sub>12</sub>
1	1	1	1	C <sub>11</sub> ·C <sub>12</sub>	1	1	C <sub>11</sub> & C <sub>12</sub>

【0108】

【表5】

【0109】が得られる。つまり表5は拡張型4入力2  
出力加算器100が満足すべき入出力関係を例示する真  
理値表の第2の例である。表5の真理値表をブール式で

表現すると、

【0110】

【数8】

$$Co1 = (A \cdot B) \& (C \cdot D) + \sim(A \cdot B) \& C \& D + \sim(C \cdot D) \& A \& B$$

$$Co2 = A \& B \& (C + D) + C \& D \& (A + B)$$

$$SO = A \cdot B \cdot C \cdot D \cdot C_{11} \cdot C_{12}$$

$$CO = \sim(A \cdot B \cdot C \cdot D) \& (C_{11} \& C_{12}) + (A \cdot B \cdot C \cdot D) \& (C_{11} + C_{12})$$

【0111】となる。

【0112】図9は数8に基づいて構成された、拡張型  
4入力2出力加算器100の構成の第2例を示す回路図  
である。ゲートG17～G21を用いて下位出力SO、  
上位出力COを得る構造は図8に示された構成と同様で  
ある。

【0113】ORゲートG31には入力A、Bが与えら  
れ、その出力は入力C、Dと共にNANDゲートG33  
に与えられる。ゲートG31、G33は複合ゲートとし  
て構成することができる。ORゲートG32には入力  
C、Dが与えられ、その出力は入力A、Bと共にNAN  
DゲートG34に与えられる。ゲートG32、G34は  
複合ゲートとして構成することができる。NANDゲ  
ートG33、G34の出力は共にNANDゲートG35に  
与えられ、NANDゲートG35は第2の桁上げ出力C  
<sub>0</sub>2を出力する。

【0114】NANDゲートG36には入力A、Bが与  
えられ、NANDゲートG37には入力C、Dが与えら  
れる。NANDゲートG36の出力はXORゲートG1  
8の出力と共にNORゲートG38に与えられる。NA

NDゲート37の出力はXORゲートG17の出力と共  
にNORゲートG39に与えられる。ANDゲートG4  
0にはXORゲートG17、G18の出力が与えられ  
る。NORゲートG38、G39、ANDゲートG40  
の出力はNORゲートG41に与えられ、その出力はイ  
ンバータG42に与えられ、インバータG42は第1の  
桁上げ出力C<sub>0</sub>1を出力する。ゲートG40、G41は  
複合ゲートとして構成することができる。

【0115】表4と表5とを比較して分かるように、こ  
れらが異なるのは第1及び第2の桁上げ出力C<sub>0</sub>1、C  
<sub>0</sub>2が入れ替わっている箇所が存在するのみである。つ  
まり下位出力SOと上位出力COに関しては異なってい  
ない。下位出力SOと上位出力COはいずれも第1及び  
第2の桁上げ入力の論理積、論理和、排他的論理和の関  
数であって第1及び第2の桁上げ入力C<sub>1</sub>1とC<sub>1</sub>2に  
関して可換である。

【0116】そして

【0117】

【数9】

21

$$\begin{aligned}
 Co1 \& Co2 &= A \& B \& C \& D \\
 Co1 \mid Co2 &= (A \mid B) \& (C \mid D) \mid (A \& B \mid C \& D) \\
 Co1 \sim Co2 &= Co1 \& \sim Co2 \mid \sim Co1 \& Co2 \\
 &= \sim (A \& B) \& C \& D \mid A \& B \& \sim (C \& D) \mid (A \sim B) \& (C \sim D)
 \end{aligned}$$

【0118】が成立する。

【0119】つまり、図5に示された拡張型4入力2出力加算器100の構成としては、図8及び図9に示されるものに限らず、以下のブール式を満たしさえすれば良

$$\begin{aligned}
 Co1 \& Co2 &= A \& B \& C \& D \\
 Co1 \mid Co2 &= (A \mid B) \& (C \mid D) \mid (A \& B \mid C \& D) \\
 Co1 \sim Co2 &= \sim (A \& B) \& C \& D \mid A \& B \& \sim (C \& D) \mid (A \sim B) \& (C \sim D) \\
 S0 &= A \sim B \sim C \sim D \mid Ci1 \sim Ci2 \\
 Co \sim &= (A \sim B \sim C \sim D) \& (Ci1 \& Ci2) \mid (A \sim B \sim C \sim D) \& (Ci1 \mid Ci2)
 \end{aligned}$$

【0121】図8に示された回路や図9に示された回路のいずれの場合も、拡張型4入力2出力加算器100のクリティカルパスはG17（或いはG18）、G19、G21を通る経路ではない。第1及び第2桁上げ入力Ci1、Ci2は、XORゲートG17、G18の出力が確定した後に確定するためである。

【0122】第1及び第2桁上げ入力Ci1、Ci2は、その一つ低いビット位置からの第1及び第2の桁上げ出力Co1、Co2が用いられるので、これらが確定するのに必要な時間を見積もる必要がある。図8に示された回路及び図9に示された回路において、第1及び第2の桁上げ出力Co1、Co2を確定するのに必要なゲート段数はそれぞれ2段及び4段である（図8に示された回路においては図9に示された回路と比較して、第1及び第2の桁上げ出力Co1、Co2を得るのに必要なゲート段数は少なくて済む。）。既述のように一般的にはXOR1段の遅延が他の論理ゲートの遅延の1段分よりも大きく、2段分程度であることを考慮すると、第1及び第2の桁上げ出力Co1、Co2を確定するのに必要な遅延時間はXOR2段分よりも少ない。

【0123】下位出力のS0を得るためには第1及び第2の桁上げ出力Co1、Co2が確定してから更にXORゲートG20、G21の2段分の遅延時間が必要であり、これは結局XOR3段分とXOR4段分の間となる。しかし拡張型4入力2出力加算器100は4入力2出力加算器200よりXOR1段分も遅れはしない。

【0124】図1乃至図4に示された加算ブロックのうち、Tree回路の1段目に属するもの（拡張型4入力2出力加算ブロック1a、4入力2出力加算ブロック2a～2c）、2段目に属するもの（4入力2出力加算ブロック2d、2e）、3段目に属するもの（4入力2出力加算ブロック2f）の順番に加算処理が行われる。従って、部分積の第1要素pp<sub>0</sub>～pp<sub>15</sub>及び第2要素pc<sub>0</sub>～pc<sub>15</sub>が確定してから、2つの最終中間和となる、

22

い。つまり図5中の拡張型4入力2出力加算器100が各々異なる構成を有していても良い。

【0120】

【数10】

4入力2出力加算ブロック2fの下位出力s<sub>0</sub>、及び上位出力c<sub>0</sub>が確定するまでの遅延時間は、拡張型4入力2出力加算ブロック1段と、4入力2出力加算ブロック2段を経由するので、XORゲートに換算して(3+α)+2×3=9+α（但し0<α<1）段分となる。

【0125】つまり、遅延時間はXOR10段分も必要ではなく、従来の技術に比べて高速化が図られている。しかも加算ブロックの数は7個で足りており、回路規模も抑制できるという効果がある。既述のように、本実施の形態では拡張型4入力2出力加算器100を従来の4入力2出力加算器200とビット毎に置換するのみで足りる。しかも加算ブロック全体が35個の加算器で構成されるのに対して、わずか5個の加算器を置換するのみでよい。

【0126】図8と図27を比較すると、拡張型4入力2出力加算器100と4入力2出力加算器200の回路規模の比は1.5:1程度であるので、上記置換によって拡張型4入力2出力加算ブロック1aが4入力2出力加算ブロック22aよりも回路規模が増大してもそれは乗算器全体からみれば無視できる程度である。

【0127】拡張型4入力2出力加算ブロック1a以外は、全て従来の技術を用いて本実施の形態を構成することができる。図2乃至図4を用いて詳細に説明すると、4入力2出力加算ブロック2aには部分積P<sub>4</sub>の第1要素pp<sub>4</sub><40:8>のうち第10ビット目～第40ビット目と、部分積P<sub>5</sub>の第1要素pp<sub>5</sub><42:10>の全てと、部分積P<sub>6</sub>の第2要素pc<sub>6</sub>と、部分積P<sub>7</sub>の第1要素pp<sub>7</sub><44:12>の全てと、部分積P<sub>8</sub>の第2要素pc<sub>8</sub>と、部分積P<sub>9</sub>の第1要素pp<sub>9</sub><46:14>のうち第14ビット目～第44ビット目とが、ビット位置を整合させて与えられる。

【0128】部分積P<sub>5</sub>の第2要素pc<sub>5</sub>と、部分積P<sub>6</sub>の第2要素pc<sub>6</sub>とは、部分積P<sub>7</sub>の第1要素pp<sub>7</sub>の疑似的な下位として扱われる。

【0129】部分積 $P_i$ の第1要素 $p_{pi}$ の第8ビット目及び第9ビット目は、これらが占めるビット位置には4入力2出力加算ブロック2aが対応しないので、4入力2出力加算ブロック2aの上位出力 $co_2 < 45 : 11 >$ の疑似的な下位として扱われ、4入力2出力加算ブロック2dへと送られる。

【0130】第2要素 $p_{ci}$ は、これが占めるビット位置には4入力2出力加算ブロック2aが対応しないので、4入力2出力加算ブロック2dに送られる。

【0131】部分積 $P_i$ の第1要素 $p_{pi}$ の第45ビット目は、これが占めるビット位置には4入力2出力加算ブロック2aが対応しないので、4入力2出力加算ブロック2aの下位出力 $so_2 < 44 : 10 >$ の疑似的な上位として扱われ、4入力2出力加算ブロック2dへと送られる。第1要素 $p_{pi}$ の第46ビット目は、これが占めるビット位置には4入力2出力加算ブロック2a、2dのいずれもが対応しないので、4入力2出力加算ブロック2dの下位出力 $so_3 < 45 : 6 >$ の疑似的な上位として扱われ、4入力2出力加算ブロック2fへと送られる。

【0132】部分積 $P_i$ の第2要素 $p_{ci}$ は、これが占めるビット位置にデータが既に4つ存在するので、拡張型4入力2出力加算ブロック2aにおいて加算されはしない。これは4入力2出力加算ブロック2fへと送られる。

【0133】4入力2出力加算ブロック2fは全て第14ビット目以上に配置される。よって、4入力2出力加算ブロック2dの下位出力 $so_3$ の第6ビット目～第13ビット目はその疑似的な下位として扱われる下位出力 $so_1$ の第2ビット目～第5ビット目と、第1要素 $p_{pi}$ の第0ビット及び第1ビットと同様に、4入力2出力加算ブロック2fの下位出力 $so_1 < 62 : 14 >$ の疑似的な下位として扱われ、最終加算の対象となる。

【0134】同様に、4入力2出力加算ブロック2dの上位出力 $co_3$ の第7ビット目～第13ビット目はその疑似的な下位として扱われる上位出力 $co_1$ の第3ビット目～第5ビット目と、第2要素 $p_{ci}$ と同様に、4入力2出力加算ブロック2fの上位出力 $co_1 < 63 : 15 >$ の疑似的な下位として扱われ、最終加算の対象となる。

【0135】4入力2出力加算ブロック2bには部分積 $P_i$ の第1要素 $p_{pi} < 48 : 16 >$ のうち第18ビット目～第48ビット目と、部分積 $P_i$ の第1要素 $p_{pi} < 50 : 18 >$ の全てと、部分積 $P_i$ の第2要素 $p_{ci}$ と、部分積 $P_{i0}$ の第1要素 $p_{pi0} < 52 : 20 >$ の全てと、部分積 $P_{i0}$ の第2要素 $p_{ci0}$ と、部分積 $P_{i1}$ の第1要素 $p_{pi1} < 54 : 22 >$ のうち第22ビット目～第52ビット目とが、ビット位置を整合させて与えられる。

【0136】部分積 $P_i$ の第2要素 $p_{ci}$ と、部分積 $P_{i0}$ の第2要素 $p_{ci0}$ とは、部分積 $P_{i1}$ の第1要素 $p_{pi1}$

の疑似的な下位として扱われる。

【0137】部分積 $P_i$ の第1要素 $p_{pi}$ の第16ビット目と第17ビット目は、これらが占めるビット位置には4入力2出力加算ブロック2bが対応しないので、4入力2出力加算ブロック2bの上位出力 $co_3 < 53 : 19 >$ の疑似的な下位として扱われる。

【0138】第2要素 $p_{ci}$ は、これが占めるビット位置には4入力2出力加算ブロック2bが対応しないので、4入力2出力加算ブロック2fに送られる。

【0139】部分積 $P_{i1}$ の第1要素 $p_{pi1}$ の第53ビット目及び第54ビット目は、これらが占めるビット位置には4入力2出力加算ブロック2bが対応しないので、4入力2出力加算ブロック2bの下位出力 $so_3 < 52 : 18 >$ の疑似的な上位として扱われ、4入力2出力加算ブロック2eへと送られる。

【0140】部分積 $P_{i1}$ の第2要素 $p_{ci1}$ は、これが占めるビット位置にデータが既に4つ存在するので、拡張型4入力2出力加算ブロック2bにおいて加算されはしない。これは4入力2出力加算ブロック2eへと送られる。

【0141】4入力2出力加算ブロック2eは全て第22ビット目以上に配置される。よって、4入力2出力加算ブロック2bの下位出力 $so_3$ の第18ビット目～第21ビット目は4入力2出力加算ブロック2eの下位出力 $so_3 < 61 : 22 >$ の疑似的な下位として扱われ、4入力2出力加算ブロック2fに送られる。

【0142】同様に、4入力2出力加算ブロック2bの上位出力 $co_3$ の第19ビット目～第21ビット目はその疑似的な下位として扱われる第1要素 $p_{pi}$ の第16ビット目及び第17ビット目と共に、4入力2出力加算ブロック2eの上位出力 $co_3 < 62 : 23 >$ の疑似的な下位として扱われ、4入力2出力加算ブロック2fに送られる。

【0143】4入力2出力加算ブロック2cには部分積 $P_{i2}$ の第1要素 $p_{pi2} < 56 : 24 >$ のうち第26ビット目～第56ビット目と、部分積 $P_{i3}$ の第1要素 $p_{pi3} < 58 : 26 >$ の全てと、部分積 $P_{i3}$ の第2要素 $p_{ci3}$ と、部分積 $P_{i4}$ の第1要素 $p_{pi4} < 60 : 28 >$ の全てと、部分積 $P_{i4}$ の第2要素 $p_{ci4}$ と、部分積 $P_{i5}$ の第1要素 $p_{pi5} < 62 : 30 >$ のうち第30ビット目～第60ビット目とが、ビット位置を整合させて与えられる。

【0144】部分積 $P_{i3}$ の第2要素 $p_{ci3}$ と、部分積 $P_{i4}$ の第2要素 $p_{ci4}$ とは、部分積 $P_{i5}$ の第1要素 $p_{pi5}$ の疑似的な下位として扱われる。

【0145】部分積 $P_{i2}$ の第1要素 $p_{pi2}$ の第24ビット目と第25ビット目は、これらが占めるビット位置には4入力2出力加算ブロック2cが対応しないので、4入力2出力加算ブロック2cの上位出力 $co_3 < 61 : 27 >$ の疑似的な下位として扱われる。

【0146】第2要素 $p_{ci2}$ は、これが占めるビット位

置には4入力2出力加算ブロック2cが対応しないので、そのまま4入力2出力加算ブロック2eに送られる。

【0147】部分積P<sub>15</sub>の第1要素p<sub>p15</sub>の第61ビット目は、これが占めるビット位置には4入力2出力加算ブロック2cが対応しないので、4入力2出力加算ブロック2cの下位出力s<sub>0</sub>、＜60:26＞の疑似的な上位として扱われ、4入力2出力加算ブロック2eへと送られる。また部分積P<sub>15</sub>の第1要素p<sub>p15</sub>の第62ビット目は、これが占めるビット位置には4入力2出力加算ブロック2cが対応しないので、4入力2出力加算ブロック2eの下位出力s<sub>0</sub>、＜61:22＞の疑似的な上位として扱われ、4入力2出力加算ブロック2fへと送られる。

【0148】4入力2出力加算ブロック2fは4入力2出力加算ブロック2eの上位出力c<sub>0</sub>、下位出力s<sub>0</sub>の全てと、4入力2出力加算ブロック2dの上位出力c<sub>0</sub>、下位出力s<sub>0</sub>のうちの第14ビット目以上の部分と、4入力2出力加算ブロック2bの上位出力c<sub>0</sub>、下位出力s<sub>0</sub>のうちの第21ビット目以下の部分と、第2要素p<sub>c1</sub>、p<sub>c5</sub>との加算を行う。かかる加算において、同一ビット位置においてその入力数が4以下であることは図から明白である。

【0149】以上の様にビット位置を描いて演算するのであるから、遅延時間の面のみから考えると第2要素p<sub>c15</sub>はそのビット位置（第30ビット目）さえ同じであれば、第2要素p<sub>c15</sub>を入力する加算ブロックは7つの加算ブロックのうちどこでも良い。例えば拡張型4入力2出力加算ブロック1aを4入力2出力加算ブロックにして、4入力2出力加算ブロック2aを拡張型4入力2出力加算ブロックにすることもできる。

【0150】しかし、既述のように、拡張型4入力2出力加算ブロックにおいて配置されるべき拡張型4入力2出力加算器の個数（ビット幅）が増加するため、回路規模は増加するという短所が無視できなくなる。

【0151】よって拡張型4入力2出力加算ブロックを構成する際の拡張型4入力2出力加算器のビット幅の最小値は、第2要素p<sub>c15</sub>のビット位置（ここでは第30ビット目）（第1の原則）と、複数ビット幅の4つの入

$$Coa = (A \mid B) \& (C \mid D) \mid (A \& B \mid C \& D)$$

$$Cob = \sim (A \& B \& C \& D)$$

$$SO = A \cdot B \cdot C \cdot D \cdot (Cia \& Cib)$$

$$CO = \sim (A \cdot B \cdot C \cdot D) \& \sim Cib \mid (A \cdot B \cdot C \cdot D) \& Cia$$

【0160】となる。

【0161】拡張型4入力2出力加算器同士の間ではかかる疑似的な桁上げを行えば足りる。表6は拡張型4入力2出力加算器が満足すべき入出力関係を例示する真理

カデータのうちの最上位ビットが最も小さいもの（ここではp<sub>p0</sub>）の最上位ビットの位置（ここでは第32ビット目）と（第2の原則）、「1加算法」で必要なビット数（ここでは2。これはブースアルゴリズムの回数に依存する。第1及び第2の原則に対する補正）で決まる。

【0152】換言すれば、最も面積効率良く本実施の形態を実現するには、2次のブースアルゴリズムによって生成される第2要素p<sub>c1</sub>のうちの最も上位のビット位置のものを、第1要素p<sub>p1</sub>のうちの最上位ビットの位置が最も小さいものと、同一の拡張型4入力2出力加算ブロックに入力すれば良い。

【0153】実施の形態2. 数10から分かるように、拡張型4入力2出力加算器100間で伝達される第1及び第2の桁上げ出力C<sub>01</sub>、C<sub>02</sub>はそれら自身の値に桁上げの意味を持たせる必要はない。第1及び第2の桁上げ出力C<sub>01</sub>、C<sub>02</sub>の論理積、論理和、排他的論理和（或いはその反転）が拡張型4入力2出力加算器100間で伝達されればよい。

【0154】このことに鑑みれば、拡張型4入力2出力加算器はもっと簡単な構成で実現することができる。具体的には第1及び第2の桁上げ出力C<sub>01</sub>、C<sub>02</sub>の代わりに新たな第1及び第2の疑似桁上げ出力C<sub>0a</sub>、C<sub>0b</sub>を導入する。そしてこれらはその隣接する上位の拡張型4入力2出力加算器の第1及び第2の疑似桁上げ出力C<sub>1a</sub>、C<sub>1b</sub>として機能する。例えば

【0155】

【数11】

$$Coa = Co1 \mid Co2, Cob = \sim (Co1 \& Co2)$$

【0156】として第1及び第2の疑似桁上げ出力C<sub>0a</sub>、C<sub>0b</sub>を設定すれば

【0157】

【数12】

$$Coa \& Cob = Co1 \cdot Co2$$

【0158】が成立する。よって第1及び第2の疑似桁上げ出力C<sub>0a</sub>、C<sub>0b</sub>、第1及び第2の疑似桁上げ出力C<sub>1a</sub>、C<sub>1b</sub>を用いて数10を書き換えれば

【0159】

【数13】

値表の例である。

【0162】

【表6】

A	B	C	D	SO	Coa	Cob	CO
0	0	0	0	Cia & Cib	0	1	~Cib
0	0	0	1	~(Cia & Cib)	0	1	Cia
0	0	1	0	~(Cia & Cib)	0	1	Cia
0	0	1	1	Cia & Cib	1	1	~Cib
0	1	0	0	~(Cia & Cib)	0	1	Cia
0	1	0	1	Cia & Cib	1	1	~Cib
0	1	1	0	Cia & Cib	1	1	~Cib
0	1	1	1	~(Cia & Cib)	1	1	Cia
1	0	0	0	~(Cia & Cib)	0	1	Cia
1	0	0	1	Cia & Cib	1	1	~Cib
1	0	1	0	Cia & Cib	1	1	~Cib
1	0	1	1	~(Cia & Cib)	1	1	Cia
1	1	0	0	Cia & Cib	1	1	~Cib
1	1	0	1	~(Cia & Cib)	1	1	Cia
1	1	1	0	~(Cia & Cib)	1	1	Cia
1	1	1	1	Cia & Cib	1	0	~Cib

【0163】図10は表6の関係を満足する拡張型4入力2出力加算器111の構成を例示する回路図である。数13を数7と比較すれば分かるように、第1の疑似桁上げ出力Coaは第1の桁上げ出力Co1と等しく、第2の疑似桁上げ出力Cobは第2の桁上げ出力Co2の反転に等しい。従って、第1及び第2の疑似桁上げ出力Coa, Cobは図8に示されたゲートG1~G9からインバータG2を省いた構造で求めることができる。

【0164】拡張型4入力2出力加算器111の桁上げは疑似的であっても、上位出力CO及び下位出力SOを出力する。XORゲートG17には入力A, Bが与えられ、XORゲートG18には入力C, Dが与えられる。XORゲートG17, G18の出力はXORゲートG19に与えられる。第1及び第2の疑似桁上げ入力Cia, CibはNANDゲートG51に与えられる。

【0165】XNORゲートG52の入力として、NANDゲートG51の出力とXORゲートG19の出力が与えられる。XNORゲートG52は下位出力SOを出力する。

【0166】第2の疑似桁上げ入力CibはXORゲートG19の出力と共にNORゲートG13に与えられる。第1の疑似桁上げ入力CiaはXORゲートG19の出力と共にANDゲートG14に与えられる。NORゲートG13の出力はANDゲートG14の出力と共にNORゲートG15に与えられ、NORゲートG15の出力はインバータG16に与えられる。ゲートG14, G15は複合ゲートとして構成することができる。インバータG16は上位出力COを出力する。

【0167】このように構成された拡張型4入力2出力加算器111は、その有するゲート数が図8に示された拡張型4入力2出力加算器100の有するゲート数よりも4つ少なく、より簡単な構造で実現できる。

【0168】但し、拡張型4入力2出力加算器111のみを直列に接続したのみでは、第2要素pciのうち最も上位のもの(図1乃至図4に即してみればpcis)と、これが対応するビット位置よりも一つ下位のビット位置(図1乃至図4に即してみれば第29ビット目)に

配置された(通常の)4入力2出力加算器200からの桁上げ出力Ciとを正しく処理することができない。逆に、拡張型4入力2出力加算器111の第1及び第2の疑似桁上げ出力Coa, Cobは、(通常の)4入力2出力加算器200の桁上げ入力CiやD入力として用いることができない。

20 【0169】従って、拡張型4入力2出力加算器111が直列に接続されたものの上位側・下位側には、それぞれ(通常の)4入力2出力加算器200との整合を採るための別の拡張型4入力2出力加算器が必要となる。

【0170】図11は拡張型4入力2出力加算ブロック1aの構成の一部を示すブロック図であり、図5に対応している。図11は図6及び図7と相まって拡張型4入力2出力加算ブロック1aの構成を示す。換言すれば、本実施の形態にかかるTree回路は図1に示されたものと同様の構成を示しており、拡張型4入力2出力加算ブロック1aはその構成のうち図5で示された部分の構成を

30 図11で示された部分の構成に置換したものである。

【0171】図11に示された構成は、図5に示された構成のうち、拡張型4入力2出力加算器100を拡張型4入力2出力加算器110~112で置換した構成を有している。詳細に言えば、第30ビット目には拡張型4入力2出力加算器110が、第31ビット目~第33ビット目には拡張型4入力2出力加算器111が、第33ビット目には拡張型4入力2出力加算器112が、それぞれ拡張型4入力2出力加算器100に代わって配置さ

40 れている。

【0172】拡張型4入力2出力加算器110のE入力にはpcisが入力され、桁上げ入力Ciとして第29ビット目の4入力2出力加算器200の桁上げ出力Coが与えられる。

【0173】第35ビット目を占める部分積はpp2, pp3の2つであり、更に「1加算法」によって加算すべき“1”が存在するので、この位置にある4入力2出力加算器200のD入力に余っている。そこでこのD入力へと拡張型4入力2出力加算器112の第1の桁上げ出力Co1が与えられ、第2の桁上げ出力Co2は同じ

50



4入力2出力加算器200の桁上げ入力Ciとして入力される。

【0174】勿論、拡張型4入力2出力加算器110のE入力に桁上げ出力Coを、桁上げ入力Ciとして第2要素pCiを各々入力してもよい。また拡張型4入力2出力加算器112の第2の桁上げ出力Co2を第35ビット目の4入力2出力加算器200のD入力として与え、拡張型4入力2出力加算器112の第1の桁上げ出力Co1を第35ビット目の4入力2出力加算器200の桁上げ入力Ciとして与えてもよい。但し、拡張型4入力2出力加算器110～112の間の接続を入れ換えることはできない。

【0175】このようにして配置された拡張型4入力2出力加算器110～112が図5に示された拡張型4入

A	B	C	D	SO	Coa	Cob	CO
0	0	0	0	$B \cdot Ci$	0	1	$E \& Ci$
0	0	0	1	$\sim(E \cdot Ci)$	0	1	$E \cdot Ci$
0	0	1	0	$\sim(E \cdot Ci)$	0	1	$E \cdot Ci$
0	0	1	1	$E \cdot Ci$	1	1	$E \& Ci$
0	1	0	0	$\sim(E \cdot Ci)$	0	1	$E \cdot Ci$
0	1	0	1	$E \cdot Ci$	1	1	$E \& Ci$
0	1	1	0	$E \cdot Ci$	1	1	$E \& Ci$
0	1	1	1	$\sim(E \cdot Ci)$	1	1	$E \cdot Ci$
1	0	0	0	$\sim(E \cdot Ci)$	0	1	$E \cdot Ci$
1	0	0	1	$E \cdot Ci$	1	1	$E \& Ci$
1	0	1	0	$E \cdot Ci$	1	1	$E \& Ci$
1	0	1	1	$\sim(E \cdot Ci)$	1	1	$E \cdot Ci$
1	1	0	0	$E \cdot Ci$	1	1	$E \& Ci$
1	1	0	1	$\sim(E \cdot Ci)$	1	1	$E \cdot Ci$
1	1	1	0	$\sim(E \cdot Ci)$	1	1	$E \cdot Ci$
1	1	1	1	$E \cdot Ci$	1	0	$E \& Ci$

【0178】

$$\begin{aligned} Coa &= (A \cdot B) \& (C \cdot D) \mid (A \& B \mid C \& D) \\ Cob &= \sim(A \& B \& C \& D) \\ SO &= A \cdot B \cdot C \cdot D \cdot E \cdot Ci \\ CO &= \sim(A \cdot B \cdot C \cdot D) \& (E \& Ci) \mid (A \cdot B \cdot C \cdot D) \& (E \cdot Ci) \end{aligned}$$

【0179】既述のように第1及び第2の疑似桁上げ出力Coa、Cobは図8に示されたゲートG1～G9からインバータG2を省いた構造で求めることができる。また、E入力も桁上げ入力Ciも、実施の形態1で示された第1及び第2の桁上げ入力Ci1、Ci2と同じ信号の意味を有している。このため上位出力CO、下位出力SOは図8に示されたゲートG10～G21を用いて得ることができる。従って、拡張型4入力2出力加算器110も拡張型4入力2出力加算器100よりも少ない

力2出力加算器100と同一機能をもつためには、拡張型4入力2出力加算器110が入力A～D及び桁上げ入力Ci並びにE入力から第1及び第2の疑似桁上げ出力Coa、Cobを出力し、拡張型4入力2出力加算器112が入力A～D及び第1及び第2の疑似桁上げ入力Ci1、Ci2から第1及び第2の桁上げ出力Co1、Co2を出力すればよい。

【0176】表7は、拡張型4入力2出力加算器110の機能を示す真理値表であり、数14は表7を満足するブール式であり、図12は表7、数14を満足する拡張型4入力2出力加算器110の構成を例示する回路図である。

【0177】

【表7】

【数14】

ゲート数で実現することができる。

【0180】表8は、拡張型4入力2出力加算器112の機能を示す真理値表であり、数15は表8を満足するブール式であり、図13は表8、数15を満足する拡張型4入力2出力加算器112の構成を例示する回路図である。

40 【0181】

【表8】

31

A	B	C	D	S0	Co1	Co2	Co
0	0	0	0	Cia & Cib	0	0	~Cib
0	0	0	1	~(Cia & Cib)	0	0	Cia
0	0	1	0	~(Cia & Cib)	0	0	Cia
0	0	1	1	Cia & Cib	1	0	~Cib
0	1	0	0	~(Cia & Cib)	0	0	Cia
0	1	0	1	Cia & Cib	1	0	~Cib
0	1	1	0	Cia & Cib	1	0	~Cib
0	1	1	1	~(Cia & Cib)	1	0	Cia
1	0	0	0	~(Cia & Cib)	0	0	Cia
1	0	0	1	Cia & Cib	1	0	~Cib
1	0	1	0	Cia & Cib	1	0	~Cib
1	0	1	1	~(Cia & Cib)	1	0	Cia
1	1	0	0	Cia & Cib	1	0	~Cib
1	1	0	1	~(Cia & Cib)	1	0	Cia
1	1	1	0	~(Cia & Cib)	1	0	Cia
1	1	1	1	Cia & Cib	1	1	~Cib

32

【0182】

【数15】

$Co1 = (A \mid B) \& (C \mid D) \mid (A \& B \mid C \& D)$   
 $Co2 = A \& B \& C \& D$   
 $S0 = A \cdot B \cdot C \cdot D \cdot (Cia \& Cib)$   
 $Co = \sim(A \cdot B \cdot C \cdot D) \& \sim Cib \mid (A \cdot B \cdot C \cdot D) \& Cia$

【0183】拡張型4入力2出力加算器112は第1及び第2の桁上げ出力Co1、Co2を出力する必要があるため、図8と同様に接続されたゲートG1～G9を必要とする。また上位出力Co、下位出力S0は拡張型4入力2出力加算器111と同様にゲートG13～G19、G41、G42で求めることができる。従って、拡張型4入力2出力加算器110も拡張型4入力2出力加算器100よりも少ないゲート数で実現することができる。

【0184】本実施の形態では拡張型4入力2出力加算器の間を伝搬する論理を最適化することにより、上記の様に回路規模を削減しているが、実施の形態1の場合より遅延時間も短縮されている。

【0185】図10、図12、図13を参照して分かるように、本実施の形態による拡張型4入力2出力加算器110～112において、自身よりも一つ下位のビット位置の拡張型4入力2出力加算器における入力A～Dの値が確定してから、自身の下位出力S0が確定するまでに必要な遅延時間は、実施の形態1に示された拡張型4入力2出力加算器100におけるそれよりも短縮される。これは実施の形態1においてXORゲートG20が必要であったのに対し、本実施の形態においてはXORゲートよりも遅延時間が短いNANDゲートG51を備えれば良いためである。従って、本実施の形態によれば拡張型4入力2出力加算ブロック1aにおける遅延時間はほぼXOR3段分と見積もることができる。

【0186】本実施の形態を用いてTree回路を構成した場合、その1段目に属するもの（拡張型4入力2出力加算ブロック1a、4入力2出力加算ブロック2a～2c）、2段目に属するもの（4入力2出力加算ブロック2d、2e）、3段目に属するもの（4入力2出力加算

20 ブロック2f）の順番に加算処理が行われる。従って、部分積の第1要素pp<sub>0</sub>～pp<sub>15</sub>及び第2要素pc<sub>0</sub>～pc<sub>15</sub>が確定してから、2つの最終中間和となる、4入力2出力加算ブロック2fの下位出力s<sub>07</sub>及び上位出力c<sub>07</sub>が確定するまでの遅延時間は、拡張型4入力2出力加算ブロック1段と、4入力2出力加算ブロック2段を経由するので、XORゲートに換算して3+2×3=9段分となる。これは実施の形態1で実現された(9+α)段分(0<α<1)よりも短い。

30 【0187】実施の形態3. 図14は本発明の実施の形態3にかかる乗算器の構成の一部を示すブロック図である。乗数、被乗数は共に24ビットで符号付き2の補数表現され、2次のブースのアルゴリズムを用いて12個の部分積P<sub>0</sub>～P<sub>11</sub>が得られる。ここではこれらの部分積を生成する部分は示さず、部分積を加算する際に、中間和の数を順次段階的に減少させ、最終的に2つの最終中間和を生成するTree回路の概略を示している。従来技術において説明されたように数5に基づいて、部分積P<sub>j</sub>は25ビット幅の第1要素pp<sub>j</sub>と、1ビット幅の第2要素pc<sub>j</sub>と、これらの最下位ビットの位置を示す2jとによって決定される。

40 【0188】実施の形態3において示されたTree回路は、複数ビット幅の3つの入力データと1ビットの入力データとを同時に加算する回路ブロック（以降「拡張型3入力2出力加算ブロック」と称する）13aと、3入力2出力加算ブロック14a～14cと、4入力2出力加算器ブロック12a～12cとから構成されている。

50 【0189】拡張型3入力2出力加算ブロック13aは、部分積の第2要素pc<sub>11</sub>と、部分積の第1要素pp<sub>0</sub>～pp<sub>2</sub>を入力し、中間和として上位出力c<sub>011</sub>及び下位出力s<sub>011</sub>を出力する。3入力2出力加算ブロック

14 aは部分積の第1要素 $p p_3 \sim p p_6$ を入力し、中間和として上位出力 $c o_{12}$ 及び下位出力 $s o_{12}$ を出力する。3入力2出力加算ブロック14 bは部分積の第1要素 $p p_6 \sim p p_9$ を入力し、中間和として上位出力 $c o_{13}$ 及び下位出力 $s o_{13}$ を出力する。3入力2出力加算ブロック14 cは部分積の第1要素 $p p_9 \sim p p_{11}$ を入力し、中間和として上位出力 $c o_{14}$ 及び下位出力 $s o_{14}$ を出力する。

【0190】4入力2出力加算ブロック12 aは上位出力 $c o_{11}$ 、 $c o_{12}$ 及び下位出力 $s o_{11}$ 、 $s o_{12}$ を入力し、中間和として上位出力 $c o_{15}$ 及び下位出力 $s o_{15}$ を出力する。4入力2出力加算ブロック12 bは上位出力 $c o_{13}$ 、 $c o_{14}$ 及び下位出力 $s o_{13}$ 、 $s o_{14}$ を入力し、中間和として上位出力 $c o_{16}$ 及び下位出力 $s o_{16}$ を出力する。4入力2出力加算ブロック12 cは上位出力 $c o_{15}$ 、 $c o_{16}$ 及び下位出力 $s o_{15}$ 、 $s o_{16}$ を入力し、最終中間和として上位出力 $c o_{17}$ 及び下位出力 $s o_{17}$ を出力する。下位出力 $s o_{17}$ と上位出力 $c o_{17}$ とは、図示しない最終加算ブロックによって最終加算されて乗算結果が得られる。

【0191】図15及び図16は両者相まって拡張型3入力2出力加算ブロック13 aの構成を例示するブロック図である。図15及び図16は仮想線 $Q_{15}$ 、 $Q_{16}$ において連続する。

【0192】拡張型3入力2出力加算ブロック13 aは26ビットの加算を並列して行うため、1ビットの拡張型3入力2出力加算器300が第22ビット目～第26ビット目において5個配置され、1ビットの3入力2出力加算器400が第2ビット目～第21ビット目において20個、第27ビット目において1個配置されている。

【0193】3つの1ビット入力A、B、Cの総和は高々10進数でいう3迄であり、これは2つの1ビット出力SO、COで表すことができる。従って、第21ビット目以下においてはビット間を伝搬する桁上げが生じないので、3入力2出力加算器400同士の接続は不要となっている。

【0194】第22ビット目の拡張型3入力2出力加算器300の桁上げ入力Ciには部分積 $P_{11}$ の第2要素 $p c_{11}$ が入力されている。桁上げ入力Ciも、この第22ビット目というビット位置の重みを持つという点では3つの第1要素 $p p_0 < 2^2 >$ 、 $p p_1 < 2^2 >$ 、 $p p_2 < 2^2 >$ と同格であり、既述した第1の原則に従ったものである。

【0195】第1及び第2の原則に対する補正として「1加算法」を施すので、第24ビット目及び第25ビット目の拡張型3入力2出力加算器300のC入力に第1要素 $p p_0 < 2^4 >$ の論理反転が与えられ、第26ビット目の拡張型3入力2出力加算器300のC入力には第1要素 $p p_0 < 2^4 >$ が与えられる。この故に1ビッ

トの拡張型3入力2出力加算器300は第26ビット目まで必要となる。そして第22ビット目乃至第26ビット目のそれぞれにおいて、拡張型3入力2出力加算器300の桁上げ出力Coは上位ビットの桁上げ入力Ciとして与えられる。

【0196】第27ビット目では部分積の数が減少するため、第1要素 $p p_2 < 2^7 >$ と、「1加算法」を施すための“1”とがそれぞれ入力A、Bとして与えられる他、入力Cには第26ビット目の拡張型3入力2出力加算器300の桁上げ出力Coが与えられる。

【0197】さて、上記の拡張型3入力2出力加算器300は以下のようにして実現することができる。拡張型3入力2出力加算器300は実は4つの1ビットデータを入力し、その位置するビットに対応した下位出力SO1つと、その位置するビットよりも1つ重みの高いビットに対応した2つの出力、即ち桁上げ出力Co並びに上位出力COを得る。つまり、

【0198】

【数16】

$$A+B+C+Ci=2(Co+CO)+SO$$

20

【0199】が成立する。

【0200】一方、下位出力SOは当該拡張型3入力2出力加算器300自身が与える出力が10進法でいう偶数か奇数かを決定するのであるから、3つの入力A～Cと桁上げ入力Ciにおける“1”の個数が偶数か奇数かで値が決まる。従って、拡張型3入力2出力加算器300が満足すべき真理値表として

【0201】

【表9】

30

A	B	C	SO	(Co, CO)
0	0	0	Ci	(0, 0)
0	0	1	$\sim Ci$	(0, Ci)
0	1	0	$\sim Ci$	(0, Ci)
0	1	1	Ci	(1, 0)又は(0, 1)
1	0	0	$\sim Ci$	(0, Ci)
1	0	1	Ci	(1, 0)又は(0, 1)
1	1	0	Ci	(1, 0)又は(0, 1)
1	1	1	$\sim Ci$	(1, Ci)

【0202】が得られる。

【0203】3つの入力A、B、Cのうち“1”の数が2のときには、桁上げ出力Coと上位出力COの値は、どちらかが1でどちらかが0であれば良いので、機能としては $2^3 = 8$ 通り表現されている。表9の真理値表から、下位出力SOの論理はプール式で表現すると、

【0204】

【数17】

$$SO=A \cdot B \cdot C \cdot Ci$$

【0205】となる。

【0206】表10は、拡張型3入力2出力加算器300が満足すべき機能の一つを例示する真理値表である。

【0207】

50

【表10】

A	B	C	S0	Co	Co
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0

【0208】表10の真理値表をブール式で表現すると、

【0209】

【数18】

$$\begin{aligned} Co &= A \cdot B \cdot C \\ Co &= (A \cdot B \cdot C) \cdot Ci \\ S0 &= A \oplus B \oplus C \end{aligned}$$

【0210】となる。

【0211】また図17は表10の機能を満たす回路の一例を示す回路図である。ここで示された回路は回路規模を削減するため、および動作の高速化のために、桁上げ入力Ciと桁上げ出力Coの論理はいずれも反転させている。

【0212】ORゲートG61は入力A、Bを受け、ANDゲートG62はORゲートG61の出力と入力Cとを受け、ANDゲートG63は入力A、Bを受け、NORゲートG64はANDゲートG62、G63の出力を受け、桁上げ出力Coの反転を出力する。ゲートG61～G64は複合ゲートで構成することができる。

【0213】XORゲートG65は入力A、Bを受け、XNORゲートG67はXORゲートG65の出力と入力Cとを受け、NORゲートG68はXNORゲートG67の出力と桁上げ入力Ciの反転とを受けて、上位出力Coを出力する。

【0214】XNORゲートG66は入力Cと桁上げ入力Ciの反転とを受け、XORゲートG69はXNORゲートG66の出力とXORゲートG65の出力とを受けて下位出力S0を出力する。

【0215】拡張型3入力2出力加算器300のクリティカルパスは隣接する下位ビットのA、B、C入力からそれぞれ自身の上位出力S0までであり、遅延時間はXOR2段分とXOR3段分の間である。

【0216】図14に示されたTree回路の1段目に属するもの（拡張型3入力2出力加算ブロック13a、3入力2出力加算ブロック14a～14c）、2段目に属するもの（4入力2出力加算ブロック12a、12b）、3段目に属するもの（4入力2出力加算ブロック12c）の順番に加算処理が行われる。従って、部分積の第1要素pp<sub>0</sub>～pp<sub>n-1</sub>及び第2要素pc<sub>0</sub>～pc<sub>n-1</sub>が確定してから、2つの最終中間和となる、4入力2出力加算ブロック12cの下位出力s<sub>0i</sub>及び上位出力c<sub>0i</sub>が確定するまでの遅延時間は、拡張型3入力2出力加算

ブロック1段と、4入力2出力加算ブロック2段を経由するので、XORゲートに換算して $(2 + \alpha) + 2 \times 3 = 8 + \alpha$ （但し $0 < \alpha < 1$ ）段分となる。つまりXOR8段とXOR9段の間であり、従来の技術に比べて高速化が図られている。

【0217】実施の形態3においても、実施の形態1の場合と同様に、2次のブースアルゴリズムによって生成される第2要素pc<sub>i</sub>のうち最も上位のビット位置のものを、第1要素pp<sub>i</sub>のうち最上位ビットの位置が最も小さいものと、同一の拡張型3入力2出力加算器に入力した場合が、回路規模としては最小になる。

【0218】補足的説明。実施の形態1乃至実施の形態3においては、本発明を2次のブースのアルゴリズムを用いた32×32乗算器と24×24乗算器を具体例に採って説明してきたが、ここでは更に一般化した態様を説明する。

【0219】乗算器の2つの入力のうち、2次のブースのアルゴリズムのエンコードの対象となる方を乗数とし、他方を被乗数とする。乗数のビット幅が2nもしくは2n-1（nは2以上の整数）の場合、部分積はn個生成される。各部分積をP<sub>j</sub>（j=0～n-1）とすると、ビット幅が「被乗数のビット幅+1」の第1要素pp<sub>0</sub>と、部分積が負の場合に部分積の最下位桁に加算する1ビットの第2要素pc<sub>0</sub>とを導入すると、数3が成立する。

【0220】従来の技術において説明したように、（通常の）4入力2出力加算ブロックと（通常の）3入力2出力加算ブロックを用いて、n個の部分積を足し合わせて2つの最終中間和を出力するTree回路を構成する場合には、（n-1）個の第2要素pc<sub>i</sub>（j=0～n-2）は加算器ブロックの空いている端子に入力することができるが、第2要素pc<sub>(n-1)</sub>に対応する空き端子がない。この故に、第2要素pc<sub>(n-1)</sub>だけを単独に加算するか（例えば図18）、全ての第2要素pc<sub>i</sub>（j=0～n-1）をまとめて加算の対象にするか（例えば図28）していた。すなわち、Tree回路の入力データの数としては（n+1）個の扱いをしていた。

【0221】4入力2出力加算器ブロックと3入力2出力加算器ブロックを用いて構成するTree回路の入力データの数が $2^k \cdot 3^h$ （k=0, 1, 2, ..., h=0, 1, 2, ...）の場合のみ、Tree回路は「密」に構成できる（ここで「密」であるとは、同一段の各加算ブロックの入力データの到達時刻が揃っており、回路動作の並列度を最高の状態にすることができることを指す）。理由を以下に示す。

【0222】Tree回路を構成する加算ブロックの出力は2つであるため、「密」なTree回路の最終段は必ず4入力2出力加算ブロックである。そしてその前段は4入力2出力加算ブロックが2個であるか、3入力2出力加算ブロックが2個であるかのいずれかである。換言すれば

最終段の加算ブロックの前段の加算ブロックの入力の数  
は8か6かのいずれかである。このように最終段から逆  
に追っていくと、「密」なTree回路の入力データの数  
は $2^k \cdot 3^h$ である必要があることがわかる。

【0223】部分積の数 $n$ が $2^k \cdot 3^h$ で表される場合  
には、入力データの数 $(n+1)$ は $2^k \cdot 3^h$ で表され  
ず、従来の技術であれば「密」なTree回路が構成できな  
かった。

【0224】しかし本発明によると、拡張型4入力2出  
力加算ブロック、或いは拡張型3入力2出力加算ブロッ  
クを1つ用いることで、(通常の)4入力2出力加算ブ  
ロックや(通常の)3入力2出力加算ブロックと比較し  
て見かけ上1つ空き端子が増加し、従ってTree回路にお  
いて、同一段の加算ブロックに入力する全ての入力デー  
タの到達時刻を揃えることができる。つまり「密」なTr  
ee回路を構成でき、遅延時間の改善ができる。

【0225】勿論、部分積の数 $n$ が $2^k \cdot 3^h$ で表され  
ない場合には、その部分積の数 $n$ の固有の性質上、

「密」なTree回路を構成し得ないため、 $(n+1)$ 入力  
のTree回路を本発明に基づいて構成しても遅延時間の減  
少は図れない。

【0226】以上のように、乗数(2次のブースのアル  
ゴリズムのエンコードの対象となる入力)のビット幅が  
 $2 \cdot 2^k \cdot 3^h$ 、もしくは $2 \cdot 2^k \cdot 3^h - 1$ である場  
合に、本発明による拡張型4入力2出力加算ブロックな  
いし拡張型3入力2出力加算ブロックを用いて、2次の  
ブースのアルゴリズムを用いた乗算器の複数の部分積を  
足し合わせて2つの最終中間和を出力するTree回路を構  
成すれば、Tree回路を構成する同一段の各加算ブロック  
の入力データの到達時刻を揃え、Tree回路のクリティカ  
ルパスの論理段数を削減することができ、回路動作の並  
列度が向上し、乗算器の高速化を図ることが可能であ  
る。

【0227】特に、第2要素 $p_c$ のうち最も上位のビ  
ット位置のものを、第1要素 $p_d$ のうち最上位ビット  
の位置が最も小さいものと、同一の拡張型4入力2出力  
加算ブロックないし拡張型3入力2出力加算ブロックに  
入力することにより、面積効率が最も良くなる。

【0228】

【発明の効果】この発明のうち請求項1にかかるツリー  
回路によれば、拡張型加算ブロックにおいて、通常型加  
算ブロックの入力よりも、1ビットだけ多く入力する。  
よってこのツリー回路は、その数が2つにまで減減され  
た中間和と、この1ビットとを加えて最終中間和を得る  
為の加算ブロックを別途に必要としない。

【0229】この発明のうち請求項2にかかるツリー回  
路によれば、拡張型加算ブロックをツリー回路の初段に  
配置するので、ツリー回路の第2段目(初段から求めら  
れた中間和同士を更に加算する段階)にはタイミングが  
整合して中間和が与えられる。従って回路構成を大きく

することなく、中間和を求めるタイミングを整合させる  
ことで処理速度を向上させることができる。

【0230】拡張型加算ブロックに入力される第2要素  
のビット位置を含んだ上位においては、その他のビット  
位置における構成と比較して、規模の大きな構成が必要  
となる。この発明のうち請求項3にかかるツリー回路に  
よれば、そのビット位置が最大となるように、拡張型加  
算ブロックに入力される第2要素が選択されるので、拡  
張型加算ブロックの構成規模の増大を抑制することがで  
きる。

【0231】この発明のうち請求項4にかかるツリー回  
路によれば、加算されるべき部分積の数が最も多くなる  
(つまり入力される部分積の数と等しくなる)ビット位  
置が最も小さい加算ブロックが拡張型加算ブロックとし  
て機能する。加算されるべき部分積の数が、入力される  
部分積の数よりも小さいビット位置以上では、拡張型加  
算ブロックの構成規模を増大することなく、入力された  
第2要素の影響に対処することができる。つまり拡張型  
加算ブロックの構成のうち、これに入力される第2要素  
のビット位置以上において構成規模が大きくなるとは言  
え、その占める部分の増大を抑制することができ、ひい  
ては拡張型加算ブロックの構成規模の増大を抑制するこ  
とができる。

【0232】拡張型拡散ブロックに入力する第2要素の  
影響を受けて、特定ビットの加算器は、それよりも下位  
の加算器と比較して、上位伝搬出力が1ビット多く必要  
となり得る。この上位伝搬出力は特定ビットよりも上位  
に伝搬するので、特定ビットより上位においても、上位  
伝搬出力が1ビット多く必要となるビット位置がある。  
この発明のうち請求項5にかかるツリー回路は、特定ビ  
ット位置を含んだ上位において、上位伝搬出力の数を増  
加した拡張型加算器を備えているので、これに対応する  
ことができる。

【0233】この発明のうち請求項6にかかるツリー回  
路によれば、最も上位に配置された前記拡張型加算器  
(最上位拡張型加算器)に隣接した上位に配置された加  
算器は、桁上げ入力以外の入力において最上位拡張型加  
算器の上位伝搬出力を受けるので、通常型加算器を採用  
することができる。

【0234】この発明のうち請求項7にかかるツリー回  
路によれば、上位伝搬出力の一つを求めるのに必要な構  
成が簡単となり、拡張型加算器の構成、ひいては拡張型  
加算ブロックの構成規模が増大することを抑制すること  
ができる。

【0235】この発明のうち請求項8にかかるツリー回  
路によれば、上位伝搬出力が桁上げを意味する必要がな  
く、疑似桁上げ出力を上位に伝搬させることによって回  
路規模を簡単にして拡張型加算器を構成することができ  
る。

【0236】この発明のうち請求項9にかかるツリー回

路によれば、特定ビット位置よりも下位に配置された通常型加算器と、疑似桁上げ出力を入力する拡張型加算器との整合を採る拡張型加算器を設けるので、特定ビットよりも下位では通常の加算を行いつつ、拡張型加算器同士では疑似桁上げ出力を伝搬させることができる。

【0237】この発明のうち請求項10にかかるツリー回路によれば、疑似桁上げ出力を上位に伝搬させる拡張型加算器と、これよりも上位に配置される通常型加算器との整合を採る拡張型加算器を設けるので、拡張型加算器同士では疑似桁上げ出力を伝搬させつつ、拡張型加算器よりも上位に配置される通常型加算器では通常の加算を行うことができる。

#### 【図面の簡単な説明】

【図1】 本発明の実施の形態1の構成を示すブロック図である。

【図2】 図3及び図4と相まって図1の詳細を示すブロック図である。

【図3】 図2及び図4と相まって図1の詳細を示すブロック図である。

【図4】 図2及び図3と相まって図1の詳細を示すブロック図である。

【図5】 図6及び図7と相まって拡張型4入力2出力加算ブロック1aの構成を例示するブロック図である。

【図6】 図5及び図7と相まって拡張型4入力2出力加算ブロック1aの構成を例示するブロック図である。

【図7】 図5及び図6と相まって拡張型4入力2出力加算ブロック1aの構成を例示するブロック図である。

【図8】 拡張型4入力2出力加算器100の構成の第1例を示す回路図である。

【図9】 拡張型4入力2出力加算器100の構成の第2例を示す回路図である。

【図10】 拡張型4入力2出力加算器111の構成を例示する回路図である。

【図11】 拡張型4入力2出力加算ブロック1aの構成の一部を示すブロック図である。

【図12】 拡張型4入力2出力加算器110の構成を例示する回路図である。

【図13】 拡張型4入力2出力加算器112の構成を例示する回路図である。

【図14】 本発明の実施の形態3の構成を示すブロック図である。

【図15】 図16と相まって拡張型3入力2出力加算ブロック13aの構成を例示するブロック図である。

【図16】 図15と相まって拡張型3入力2出力加算ブロック13aの構成を例示するブロック図である。

【図17】 拡張型3入力2出力加算器300の構成を例示する回路図である。

【図18】 従来のTree回路の構造を示すブロック図である。

【図19】 図20及び図21と相まって図18の詳細を示すブロック図である。

【図20】 図19及び図21と相まって図18の詳細を示すブロック図である。

【図21】 図19及び図20と相まって図18の詳細を示すブロック図である。

【図22】 図23及び図24と相まって4入力2出力加算ブロック22aの構成を例示するブロック図である。

10 【図23】 図22及び図24と相まって4入力2出力加算ブロック22aの構成を例示するブロック図である。

【図24】 図22及び図23と相まって4入力2出力加算ブロック22aの構成を例示するブロック図である。

【図25】 3入力2出力加算器400の構成例を示す回路図である。

【図26】 4入力2出力加算器200の構成例を示すブロック図である。

20 【図27】 4入力2出力加算器200の一例を示す回路図である。

【図28】 従来のTree回路の構造を示すブロック図である。

【図29】 図30乃至図32と相まって図28の詳細を示すブロック図である。

【図30】 図29、図31及び図32と相まって図28の詳細を示すブロック図である。

【図31】 図29、図39及び図32と相まって図28の詳細を示すブロック図である。

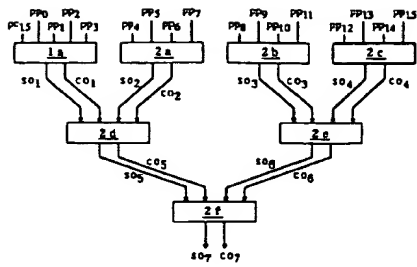
30 【図32】 図29乃至図31と相まって図28の詳細を示すブロック図である。

【図33】 従来のTree回路を例示するブロック図である。

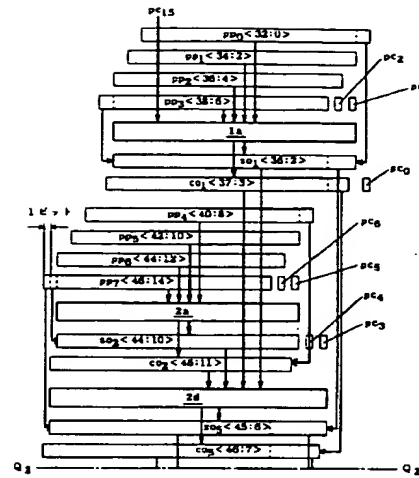
#### 【符号の説明】

$P_0 \sim P_{15}$  部分積、 $p p_0 \sim p p_{15}$  第1要素、 $p c_0 \sim p c_{15}$  第2要素、 $s o_1 \sim s o_7$ 、 $c o_1 \sim c o_7$ 、 $s o_{11} \sim s o_{17}$ 、 $c o_{11} \sim c o_{17}$ 、 $s o_{21} \sim s o_{28}$ 、 $c o_{21} \sim c o_{28}$ 、 $s o_{31} \sim s o_{46}$ 、 $c o_{31} \sim c o_{46}$  中間和、 $s o_7$ 、 $c o_7$ 、 $s o_{17}$ 、 $c o_{17}$ 、 $s o_{28}$ 、 $c o_{28}$  最終中間和、 $C o_1$  第1の桁上げ出力、 $C o_2$  第2の桁上げ出力、 $C o a$  第1の疑似桁上げ出力、 $C o b$  第2の桁上げ出力、1a 拡張型4入力2出力加算ブロック、2a $\sim$ 2f、12a $\sim$ 12c 4入力2出力加算ブロック、13a 拡張型3入力2出力加算ブロック、14a $\sim$ 14c 3入力2出力加算ブロック、100、110 $\sim$ 112 拡張型4入力2出力加算器、200 (通常の) 4入力2出力加算器、300 拡張型3入力2出力加算器、400 (通常の) 3入力2出力加算器。

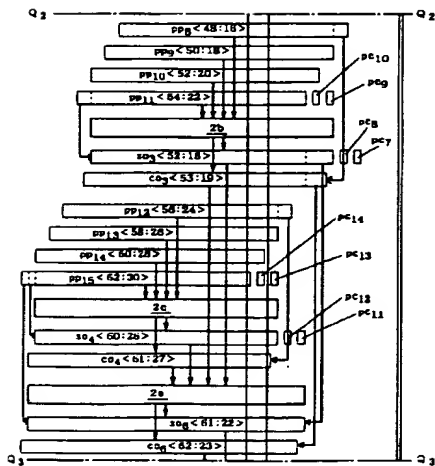
【図1】



【図2】



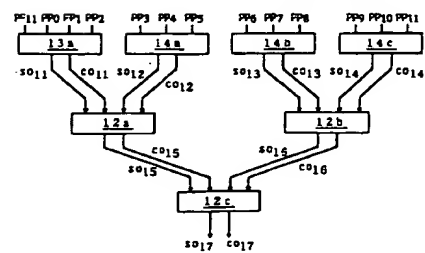
【図3】



【図4】



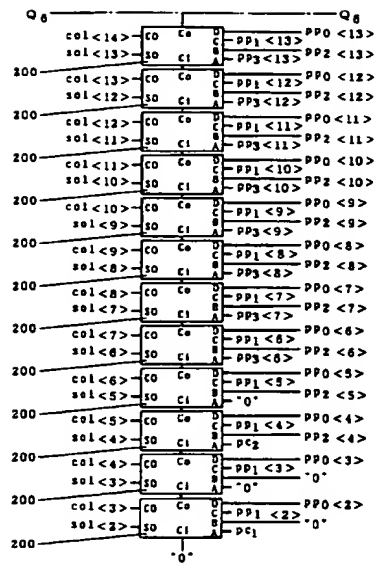
【図14】



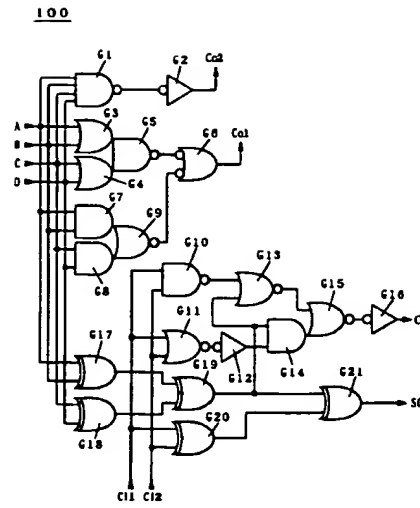




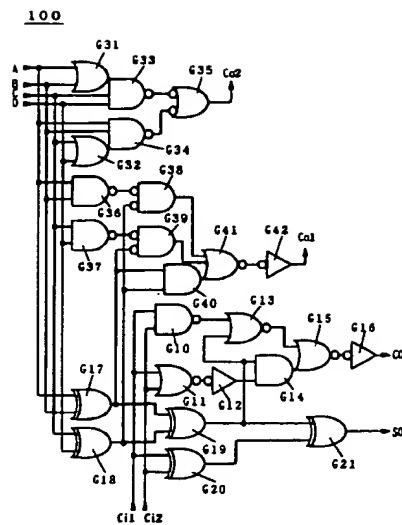
【図7】



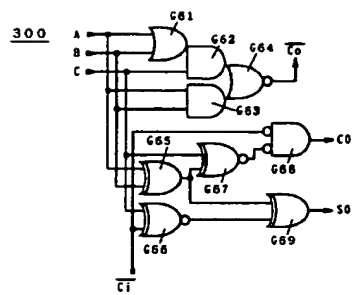
【図8】



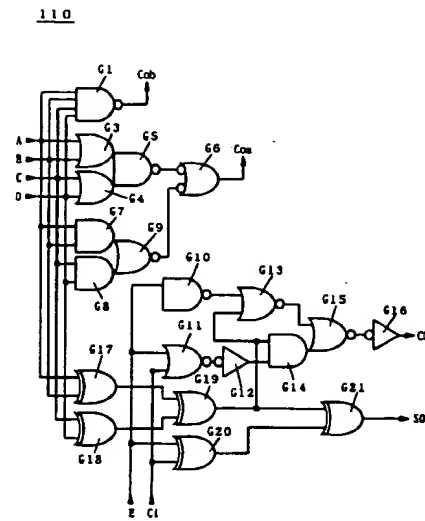
【図9】



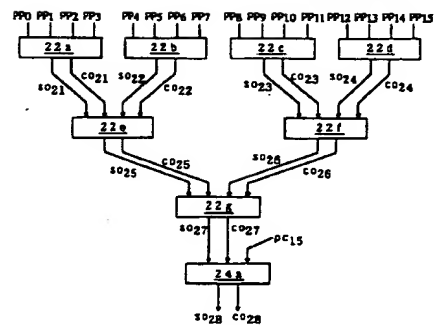
【図17】



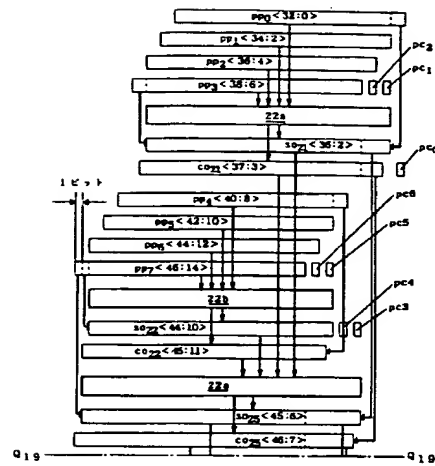
【図 12】



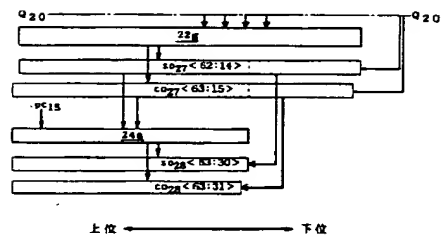
【图 18】



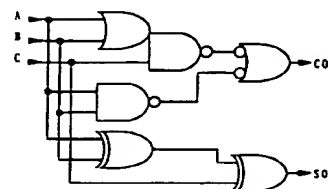
【图 19】



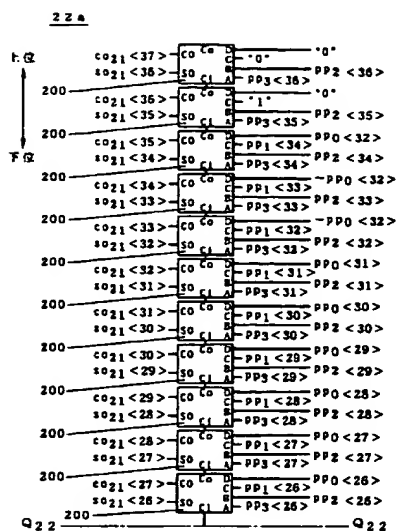
【図 20】



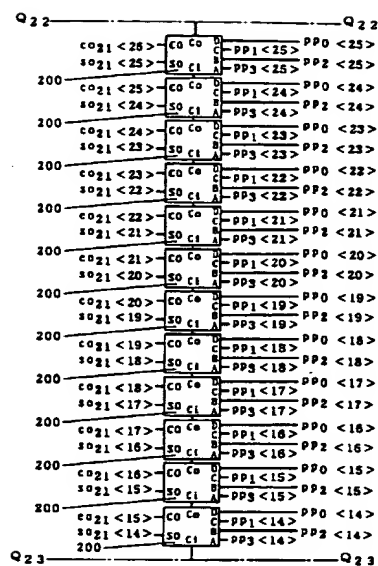
400



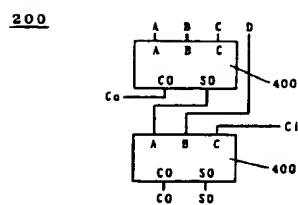
【圖 22】



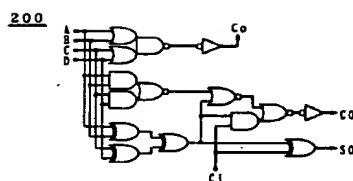
【图 2 3】



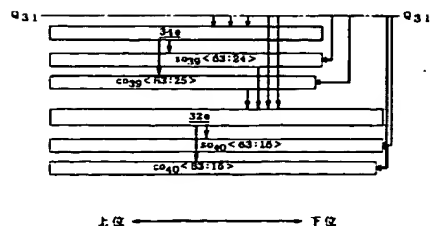
【图 26】



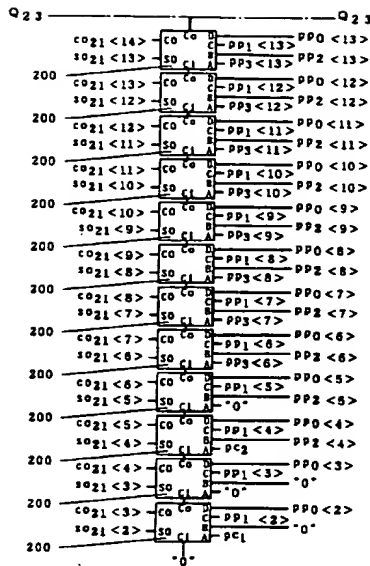
【图 27】



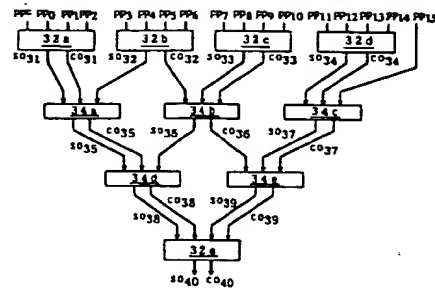
【图 3 2】



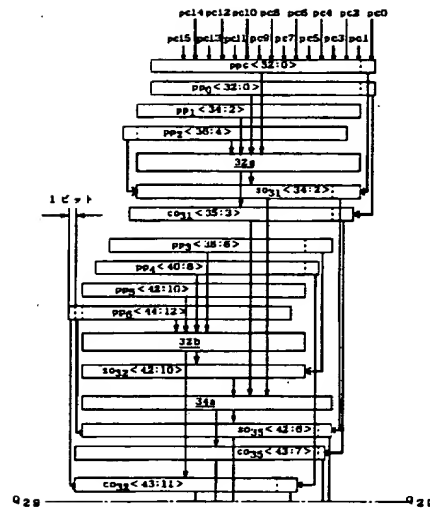
【図24】



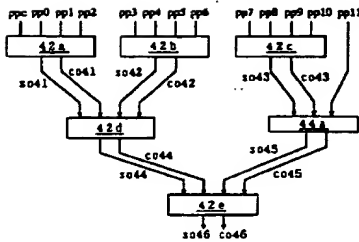
【図28】



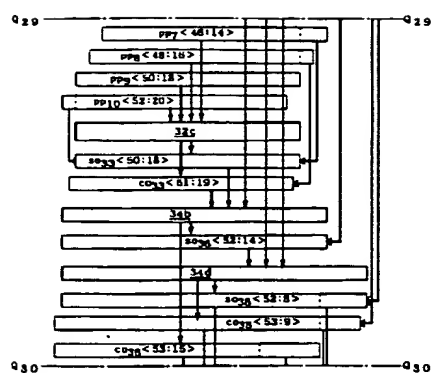
【図29】



【図33】



【図30】



【図31】

